

Electronic Theses and Dissertations, 2020-

2021

Robust and Scalable Data Representation and Analysis Leveraging Isometric Transformations and Sparsity

Alireza Zaeemzadeh
University of Central Florida

 Part of the [Electrical and Computer Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd2020>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Zaeemzadeh, Alireza, "Robust and Scalable Data Representation and Analysis Leveraging Isometric Transformations and Sparsity" (2021). *Electronic Theses and Dissertations, 2020-*. 968.
<https://stars.library.ucf.edu/etd2020/968>



ROBUST AND SCALABLE DATA REPRESENTATION AND ANALYSIS LEVERAGING
ISOMETRIC TRANSFORMATIONS AND SPARSITY

by

ALIREZA ZAEEMZADEH
B.Sc. University of Tehran, 2014
M.Sc. University of Central Florida, 2017

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2021

Major Professor: Nazanin Rahnavard

© 2021 Alireza Zaeemzadeh

ABSTRACT

The main focus of this doctoral thesis is to study the problem of robust and scalable data representation and analysis. The success of any machine learning and signal processing framework relies on how the data is *represented* and *analyzed*. Thus, in this work, we focus on three closely related problems: (i) supervised representation learning, (ii) unsupervised representation learning, and (iii) fault tolerant data analysis. For the first task, we put forward new theoretical results on why a certain family of neural networks can become extremely deep and how we can improve this scalability property in a mathematically sound manner. We further investigate how we can employ them to generate data representations that are robust to outliers and to retrieve representative subsets of huge datasets. For the second task, we will discuss two different methods, namely compressive sensing (CS) and nonnegative matrix factorization (NMF). We show that we can employ prior knowledge, such as slow variation in time, to introduce an unsupervised learning component to the traditional CS framework and to learn better compressed representations. Furthermore, we show that prior knowledge and sparsity constraint can be used in the context of NMF, not to find sparse hidden factors, but to enforce other structures, such as piece-wise continuity. Finally, for the third task, we investigate how a data analysis framework can become robust to faulty data and faulty data processors. We employ Bayesian inference and propose a scheme that can solve the CS recovery problem in an asynchronous parallel manner. Furthermore, we show how sparsity can be used to make an optimization problem robust to faulty data measurements. The methods investigated in this work have applications in different practical problems such as resource allocation in wireless networks, source localization, image/video classification, and search engines. A detailed discussion of these practical applications will be presented for each method.

This humble work is dedicated to all the hard working people
who are hunching over their keyboards right now
Ph.D. students.

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Dr Nazanin Rahnavard of the Department of Electrical Engineering at University of Central Florida. She was always keen to find out how I am proceeding and always encouraged me to keep moving forward. She consistently gave me insightful comments, while allowing this thesis to be my own work. I also want to acknowledge my thesis co-advisor Dr Mubarak Shah of the Department of Computer Science at University of Central Florida. His valuable help and comments helped me a lot during this process. It was a fantastic opportunity to work with such amazing and knowledgeable people.

I also want to extend my gratitude to Professor Deanna Needell of the Department of Mathematics at the University of California, Los Angeles. Chapter 9 of this dissertation is the result of my collaboration with her, a pioneer in the field of Compressive Sensing. I want to thank her deeply for giving me this opportunity and for her support in my professional and personal life.

I have also been fortunate to work with many smart people: Professor Ronald F. DeMara, Dr. Behzad Shahrabi, Dr. Mohsen Joneidi, Dr. Sohail Salehi Mobarakeh, Dr. Mahdi Boloursaz Mashhadi, Dr. Jamie Haddock, Dr. Niccolò Bisagno, and Nazmul Karim. I want to thank them all. I want to specially thank Mohsen Joneidi, with whom I had the most collaboration and discussions, for his valuable help and comments. I also want to extend my sincerest gratitude to Sohail Salehi Mobarakeh, who has been a great co-author, friend, and roommate throughout my studies.

I am also grateful to my parents, who have supported me unconditionally throughout my life. I owe it all to them. I am also grateful to my loving siblings, Rokhsareh and Afsaneh, who have provided me through moral and emotional support in my life. Thank you. Last but not the least, I want to thank my dear friend and wife, Toktam. She has been by my side through this journey and her presence has made my life as an immigrant graduate student much easier.

TABLE OF CONTENTS

LIST OF FIGURES	xiv
LIST OF TABLESxxiv
CHAPTER 1: INTRODUCTION	1
1.1 Supervised Representation Learning	1
1.2 Unsupervised Representation Learning	5
1.3 Fault Tolerant Data Analysis	6
CHAPTER 2: BACKGROUND	9
2.1 Deep Neural Networks	9
2.2 Compressive Sensing	12
CHAPTER 3: NORM-PRESERVATION: WHY RESIDUAL NETWORKS CAN BECOME EXTREMELY DEEP?	15
3.1 Norm-Preservation of Residual Networks	18
3.2 Procrustes Residual Network	23
3.3 Experiments	30

3.3.1	Norm-Preservation	31
3.3.2	Optimization Stability and Learning Dynamics	34
3.3.3	Classification Performance	37
3.4	Conclusions	38
CHAPTER 4: OUT-OF-DISTRIBUTION DETECTION USING UNION OF 1-DIMENSIONAL SUBSPACES		40
4.1	Related Work	41
4.2	Union of 1-dimensional Subspaces for Out-of-Distribution Detection	43
4.2.1	Enforcing the Structural Constraints	46
4.3	Out-of-distribution Detection Test	48
4.4	Experiments	50
4.5	Conclusion	55
CHAPTER 5: ITERATIVE PROJECTION AND MATCHING: FINDING STRUCTURE-PRESERVING REPRESENTATIVES AND ITS APPLICATION TO COMPUTER VISION		56
5.1	Iterative Projection and Matching (IPM)	57
5.2	Applications of IPM	60
5.2.1	Active Learning	60

5.2.2	Learning Using Representatives	63
5.2.2.1	Representatives To Generate Multi-view Images Using GAN . . .	64
5.2.2.2	Finding Representatives for UCF-101 Dataset	65
5.2.2.3	Finding Representatives for ImageNet	67
5.2.3	Video Summarization	68
5.3	Conclusions	70
CHAPTER 6: FACE IMAGE RETRIEVAL WITH ATTRIBUTE MANIPULATION . . .		71
6.1	Related Work	74
6.2	Our Approach	75
6.2.1	Extracting Orthogonal Basis Set for Disentangled Semantics	78
6.2.2	Retrieval Using Orthogonal Decomposition	81
6.3	Experiments	84
6.4	Conclusion	90
CHAPTER 7: ADAPTIVE NON-UNIFORM COMPRESSIVE SAMPLING FOR TIME- VARYING SIGNALS		91
7.1	System Model	93
7.2	Bayesian Inference of Importance Levels	94

7.3	Measurement Matrix Design	98
7.4	Numerical Experiments	99
7.4.1	Performance evaluation for sparse signals in canonical basis	100
7.4.2	Performance evaluation for sparse signals in the DCT domain	104
7.5	conclusions	106
CHAPTER 8: MISSING SPECTRUM-DATA RECOVERY IN COGNITIVE RADIO NETWORKS USING PIECEWISE CONSTANT NONNEGATIVE MATRIX FACTORIZATION		107
8.1	System Model	109
8.2	PC-NMF: Piecewise Constant Nonnegative Matrix Factorization	112
8.3	Majorization-Minimization for Piecewise Constant NMF	115
8.4	Numerical Results	118
8.5	Conclusions	122
CHAPTER 9: A BAYESIAN APPROACH FOR ASYNCHRONOUS PARALLEL SPARSE RECOVERY		123
9.1	System Model	124
9.2	Probability Model	125
9.3	Inference via Sequential Bayesian Updating	127

9.4	Numerical Experiments	129
9.4.1	Experiments in MATLAB	129
9.4.2	Experiments in C++	130
9.5	Conclusions	132
CHAPTER 10: ROBUST TARGET LOCALIZATION BASED ON SQUARED RANGE ITERATIVE REWEIGHTED LEAST SQUARES		134
10.1	System Model	136
10.2	Robust Localization From Squared Range Measurements	137
10.2.1	The Squared Range Iterative Reweighted Least Squares (SR-IRLS) Approach	141
10.2.2	The Squared Range Gradient Descent (SR-GD) Approach	144
10.3	Numerical Results	147
10.3.1	Scenario I	148
10.3.2	Scenario II	152
10.4	Conclusions	155
CHAPTER 11: CONCLUSION		156
11.1	Contributions	156

11.1.1 Analyzing and Improving the Optimization Dynamics of Deep Residual Networks	157
11.1.2 Union of Low-Dimensional Subspace for Outlier Detection Using Deep Neural Networks	158
11.1.3 Design and Analysis of a Linear-Time Subset Selection Algorithm for Large-Scale Data	158
11.1.4 Design and Analysis of Provably Convergent Optimizers for Problems with Sparsity Constraint	159
11.1.5 Bayesian Inference for Efficient and Robust Compressive Sensing	159
11.2 Future Directions	160
APPENDIX A: PROOFS FOR THEORETICAL RESULTS PRESENTED IN CHAPTER 3 162	
A.1 Proof of Theorem 3.1	163
A.2 Proof of Theorem 3.2	165
A.3 Proof for Corollary 3.1	166
APPENDIX B: IMPLEMENTATION DETAILS AND FURTHER EXPERIMENTAL RE- SULTS FOR THE METHOD PROPOSED IN CHAPTER 4 167	
B.1 Implementation Details	168
B.2 Additional Experiments	169

B.3	Related Methods: Leveraging OOD Data for OOD Detection	174
APPENDIX C: FURTHER EXPERIMENTAL RESULTS FOR THE METHOD PROPOSED		
	IN CHAPTER 5	177
C.1	Finding Representatives for ImageNet Dataset	178
C.2	Finding Representatives for UCF-101 Dataset	179
APPENDIX D: IMPLEMENTATION DETAILS AND FURTHER EXPERIMENTAL RE-		
	SULTS FOR THE METHOD PROPOSED IN CHAPTER 6	183
D.1	Implementation Details	184
D.2	Additional Experiments	185
APPENDIX E: DERIVATION OF UPDATE RULES FOR THE ALGORITHM PRESENTED		
	IN CHAPTER 9	192
E.1	Tally Score	194
E.2	Processor Reliability Score	195
E.3	Observation Reliability	195
APPENDIX F: CHOOSING HYPERPARAMETER FOR ALGORITHM PROPOSED IN		
	CHAPTER 10	197
APPENDIX G: PROOF OF THEOREM 10.3		
		200

LIST OF REFERENCES	203
------------------------------	-----

LIST OF FIGURES

2.1	An illustration of the decision boundary learned by an SVM classifier for a binary classification problem in a 2-dimensional space. The vector w is perpendicular to the decision boundary and the scalar b determines its position with respect to the origin.	10
3.1	ResNet architecture and its building blocks. Each conv block represents a sequence of batch normalization, ReLU, and convolution layers. conv* block represents the regularized convolution layer.	23
3.2	The ratio of gradient norm at output to gradient norm at input, i.e., $\ \frac{\partial \mathcal{E}}{\partial x_{l+1}}\ _2$ to $\ \frac{\partial \mathcal{E}}{\partial x_l}\ _2$, of a convolution layer for different number of input and output channels at 10 th training epoch (a) with, and (b) without the proposed regularization on the singular values of the convolution.	29
3.3	Training on CIFAR10. Gradient norm ratio over the first 100 epochs for transition blocks (blocks that change the dimension) and non-transition blocks (blocks that do not change the dimension). The darker color lines represent the transition blocks and the lighter color lines represent the non-transition blocks. The proposed regularization enhances the norm-preservation of the transition blocks effectively.	32

3.4	Loss (black lines) and error (blue lines) during training procedure on CIFAR10. Solid lines represent the test values and dotted lines represent the training values. This experiments shows how the residual connections enhance the stability of the optimization and how the proposed regularization enhances the stability even further.	36
3.5	Comparison of the parameter efficiency on CIFAR10 between ResNet and ProcResNet.	37
4.1	Overall architecture of the proposed framework. A neural network (e.g., WideResnet28) maps the input onto a feature space. Then, the cosine similarities between the extracted feature x_n and the class vectors w_l are used to compute the class membership probabilities. w_l s are set to predefined orthonormal vectors and are not updated during training. This leads to the desired embedding, union of uncorrelated 1-dimensional subspaces. At test time, the cosine similarity between the test samples and the first singular vector corresponding to each class is used to distinguish between the ID and OOD samples.	46
4.2	3-dimensional representation of the features belonging to the first 3 classes of CIFAR10 training set, extracted from WideResNet with and without the proposed embedding: (a) features extracted from a plain WideResnet, (b) features extracted after enforcing the proposed embedding, and (c) same as (b) after ℓ_2 -normalizing the feature vectors. The solid lines represent the direction of the first singular vector corresponding to each class. All the figures contain 3,000 feature vectors.	48

4.3	3-dimensional representation of the features extracted from a plain WideResNet and the same network with our proposed embedding. (a) ID features extracted from plain network, (b) OOD features extracted from plain network, (c) ID features extracted using our embedding, and (d) OOD features extracted using our embedding. The solid lines represent the direction of the first singular vector corresponding to each class. OOD samples, extracted using our embedding, have larger angular distance to their closest singular vector. All the figures contain 3000 samples.	51
4.4	(a) Empirical probability distribution of the spectral discrepancy of samples belonging to CIFAR10 (ID) and different OOD datasets. (b) Detection error for different values of critical spectral discrepancy ϕ^* . Both the spectral discrepancy histogram and the best ϕ^* do not change significantly for different datasets.	54
5.1	Multi-view face generation results for a sample subject in testing set using CR-GAN [1]. The network is trained on reduced training set (9 images per subject) using random selection (first row), K-medoids (second row), DS3 [2] (third row), and IPM (fourth row). The fifth row shows the results generated by the network trained on all the data (360 images per subject). IPM-reduced dataset generates closest results to the complete dataset.	64

5.2	t-SNE visualization [3] of two randomly selected classes of UCF-101 dataset and their representatives selected by different methods. ((a)) Decision function learned by using all the data. The goal of selection is to preserve the structure with only a few representatives. ((b)) Decision function learned by using 2 (first column), 5 (second column), and 10 (third column) representatives per class, using K-medoids (first row), DS3 [2] (second row), and IPM (third row). IPM can capture the structure of the data better using the same number of selected samples.	67
6.1	Example of face image retrieval by considering both the attribute <i>adjustment</i> and attribute <i>preference</i> specified by the user.	72
6.2	The overall architecture of the proposed face image retrieval framework. The intermediate latent space, \mathcal{W}^+ , is generated by employing StyleGAN encoder proposed in [4]. Then, the orthogonal and sparse basis vectors $\{\mathbf{f}_m\}_{m=1}^M$ are extracted using a fairly small set of face images with attribute annotations. Utilizing the basis vectors, we adjust the query, decompose the dissimilarity vectors, and assign preference to different attributes.	77
6.3	Qualitative evaluation of the learned attribute directions. In each pair of images, the image on the right is synthesized after moving the latent vector corresponding to the image on the left along an attribute direction. For attributes <code>Black Hair</code> and <code>Baldness</code> , the baseline is affecting the smile and the eyes as well, an artifact that is not present in the image manipulated by our method. For attribute <code>Mustache</code> , our method is able to add mustache to the face while not affecting the beard as much as the baseline.	82

6.4	Qualitative evaluation of face image retrieval by considering both the <i>adjustment</i> and attribute <i>preference</i> . The user is able to both adjust multiple attributes in the query face and to customize the similarity metric by assigning preference to the attributes. .	87
6.5	Impact of attribute preference on nDCG and identity similarity of the search results obtained by our method.	88
6.6	The energy concentrated in the top, most relevant, entries of the attribute vectors, averaged over all the attributes, for different values of the sparsity regularization parameter λ	89
7.1	Overall block diagram of the proposed framework. Reconstructed signal, at each time step, is utilized to generate the measurement matrix.	92
7.2	Graphical representation of the generative model.	95
7.3	(a) Support of the signal and (b) the expected value of the inferred importance levels, i.e., \bar{c}_n , for the first 30 coefficients of the signal. $M = 60$, $N = 200$, SNR = 20 dB, and $W = 5$	102
7.4	Performance of ANCS for different values of p_{01} . The total sensing energy is the same for all the methods. $N = 200$, SNR = 20 dB, and $T = 30$, and $M = 60$	102
7.5	TNMSE (in dB) for of different recovery algorithm with and without ANCS as the sampling step for $N = 200$, SNR = 20 dB, and $T = 30$	103

7.6	Performance of ℓ_1 minimization and SA-MMSE with and without ANCS as the sampling step in terms of TNMSE (in dB). of different methods for $N = 200$, $p_{01} = 0.02$, and $M = 60$	104
7.7	TNMSE (in dB) versus M of ANCS. $N = 200$, SNR = 20 dB, and $T = 30$, and $M = 60$	105
7.8	Plot of TNMSE (in dB) of ANCS for different values of fault rate. $N = 200$, SNR = 20 dB, and $T = 30$, and $M = 60$	106
8.1	The power distribution of 2 PUs (squares) and deployment of 10 SUs (triangles) without considering shadowing effect.	110
8.2	Structure of data generated by $N_P = 3$ PUs in $N_R = 3$ dimensional space.	112
8.3	Network topology consisting of 3 PUs and $N_R = 20$ SUs marked by triangles.	118
8.4	Power levels of a single SU.	120
8.5	Performance of the proposed method for different noise levels and probability of miss, averaged over 200 Monte Carlo trials.	120
8.6	Original power levels of different PUs and the estimated activation levels with PC-NMF and WNMF.	122

9.1	Comparison of the number of time steps until convergence versus the number of processors used in different methods, when (a) all processors complete an iteration in a single time step and (b) half of the processors complete an iteration every four time steps.	130
9.2	Performance of different multi-processor sparse recovery algorithm implemented using C++ programming language and OpenMP platform. Twenty percent of the processors are slow.	131
9.3	Mean convergence time of different multi-processor sparse recovery algorithms over 50 trials. Ten processors are solving the sparse recovery problem and two cores are slow.	132
10.1	Convergence of SR-IRLS, SR-GD, and the hybrid method. Labels show the execution time of different algorithms at different iterations.	149
10.2	Robustness against outliers for 200 Monte Carlo trials, $\sigma = 55$ and $R = 10$. Number of outlier sensors is set to $\beta \times R$	150
10.3	Performance of the localization methods versus number of sensors for 1000 Monte Carlo trials and $\beta = 0.4$	151
10.4	Comparison of the RMSEs in an environment with no outlier sensor, $\beta = 0$. .	152
10.5	Geometry of the sensors, marked as triangle, and the city center area, marked as gray square, in a real world operating cellular radio network.	153
10.6	Mean RMSE of different localization methods versus contamination ratio, for 100 MC trials.	154

B.1	Energy Ratio of the training samples along the first 100 singular vectors of features extracted using WideResNet with and without our proposed embedding trained on (a) CIFAR10 and (b) CIFAR100. The proposed embedding increases the energy along the first singular vector from 98.3% to 99.9% for CIFAR 10 and from 91.8% to 99.8% for CIFAR100.	169
B.2	Correlation of different singular vectors of noisy data with the same singular vector of clean data, averaged over 10 trials. Feature vectors corresponding to the first class of CIFAR10 act as the data and the feature vectors belonging to other classes are used as outliers. Noise levels up to 50% have almost no impact on the direction of the first singular vector.	170
B.3	Area Under ROC curve using the proposed framework versus the number of the Monte Carlo samples used for estimating $p(\phi_n < \phi^*)$. The networks are trained on CIFAR10 and CIFAR100 and tested on TINr as the OOD dataset.	171
B.4	ROC curves for different variants of the proposed scheme in logarithmic scale, using CIFAR10 (ID) and TINr (OOD). WideResNet (WRN) with depth of 28 and width of 10 is used as the deep feature extractor.	171
C.1	Selected images by IPM (left) and K-medoids (right) from five sample classes of ImageNet [5]. Note that the IPM-selected samples are less cluttered with other objects, making them better representatives of the class.	178

C.2	t-SNE visualization [3] of different randomly selected pairs of classes of UCF-101 dataset and their representatives selected by different methods. (Left) Decision function learned by using all the data. The goal of selection is to preserve the structure with only a few representatives. (Right) Decision function learned by using 2 (first column), 5 (second column), and 10 (third column) representatives per class, using K-medoids (first row), DS3 [2] (second row), and IPM (third row). IPM can capture the structure of the data better using the same number of selected samples.	181
C.3	Frames of the selected video clips by IPM (left) and DS3[2] (right), for a few sample classes of UCF-101 dataset[6]. Different actions are more visible and/or less cluttered, in the clip selected by IPM.	182
D.1	An example of modifying the retrieval results using continuous, real-valued, modification vector. The attribute intensity for <code>Pale Skin</code> for the query face is estimated as 0.12. The user is able to modify the results by increasing it to 0.5 and then to 1. .	186
D.2	Examples of retrieved images by our method and the compositional learning method in [7] and their corresponding nDCG and identity similarity. (a) Changing the attribute <code>Young</code> to 0, (b) Changing the attribute <code>Heavy makeup</code> to 1, and (c) Changing the attribute <code>Mouth slightly open</code> to 0. In all of these examples, our method outperforms the baseline in both the evaluation metrics. Qualitatively, the retrieved images by method can modify the attribute, while preserving the other attributes, such as skin tone, hair color, smiling, etc, better.	187

D.3	Attribute manipulation results using our method and the method proposed in [8] on two synthetic images. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. Notice the impact of adjusting attributes <code>Chubby</code> and <code>Pale skin</code> on the smile in images edited using [8].	188
D.4	Attribute manipulation results using our method and the method proposed in [8] on two images from CelebA dataset. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. The obtained directions by the baseline leads to more artifacts compared to the directions obtained by our method.	188
D.5	Ratio of ℓ_2 norm of different attribute vectors in each layer over the total ℓ_2 norm of the vector, for our method and InterFaceGAN [8]. Our method often concentrates most of the energy of the vector in a few layers. For example, vectors corresponding to <code>Bangs</code> and <code>Baldness</code> have a similar energy profile and only manipulated the layers corresponding to the coarse structures, i.e., first few layers. On the other hand, vectors corresponding to <code>Black hair</code> and <code>Pale skin</code> mainly change the last few layers, which are responsible for finer structures in the face.	189
F.1	Comparison of the IRLS weight function, its convex approximation, and the Huber norm.	199

LIST OF TABLES

3.1	Mean and maximum generalization gap (%) during the first 100 epochs of training on CIFAR10 for different network architectures, averaged over 10 runs.	35
3.2	Performance of different methods on CIFAR-10 and CIFAR-100 using moderate data augmentation (flip/translation). The modified transition blocks in ProcResNet can improve the accuracy of ResNet significantly.	38
3.3	Ablation study on ResNet with 164 layers on CIFAR100.	38
4.1	A comparison of OOD detection results, in terms of F1-score, for different ID and OOD datasets. † represents the results achieved by our re-run of the publicly available codes. The bottom section summarizes the performance of the methods that use a subset of OOD samples for hyperparameter tuning, such as finding the best perturbation magnitude. Our method does not have any parameters to be tuned. . . .	51
4.2	Performance of the proposed framework for distinguishing ID and OOD test set data for the image classification task, using a WideResnet with depth 28 and width 10. ↑ indicates larger value is better and ↓ indicates lower value is better. All the methods use the same network architecture.	52
4.3	Ablation study of the proposed framework using CIFAR10 (ID) and TINr (OOD). While enforcing the structure hurts the ID accuracy slightly, it improves the OOD detection performance significantly. The remaining two combinations, (No, Yes, No) and (No, Yes, No), are not meaningful.	53

5.1	Classification accuracy (%) for action recognition on UCF-101, at different active learning cycles. The initial training set (cycle 1) is the same for all the methods. The accuracy for cycle 1 is 54.2% and the accuracy using the full training set (9537 samples) is 82.23%.	60
5.2	Identity dissimilarities between real and generated images by network trained on reduced (using different selection methods) and complete dataset.	65
5.3	Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.	66
5.4	Top-1 classification accuracy (%) on ImageNet, using selected representatives from each class. Accuracy using all the labeled data (1.2M samples) is 46.86%. Numbers in () show the size of the selected representatives as a % of the full training set.	68
5.5	F-measure and recall scores using ROUGE-SU metric for UT Egocentric video summarization task. Results are reported for several supervised and unsupervised methods.	69
6.1	nDCG and identity similarity for different attribute-guided image retrieval methods, averaged over 1000 queries.	84
6.2	Top-5 nDCG and identity similarity for different levels of sparsity, i.e., different values of the regularization parameter λ , averaged over 1000 queries.	89
8.1	Average Running Time	121

B.1	Detection errors and f1-scores achieved by setting ϕ^* using the training set, compared to the best achievable values, on different pairs of ID and OOD datasets. . . .	172
B.2	Performance of different OOD detection tests, in term of AUROC, for distinguishing ID and OOD test set data.	173
B.3	A non-exhaustive summary of recent OOD detection methods. The information provided in the table is extracted from their corresponding manuscript or the code provided by authors.	174
C.1	Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.	179
D.1	Normalized discounted cumulative gain (nDCG) and identity similarity for the GAN-based methods using different number of training faces to obtain the attribute directions, averaged over 1000 queries. Here we are calculating the metrics on the top-5 images	190

CHAPTER 1: INTRODUCTION

Robust and scalable data representation and analysis can be succinctly described as the family of signal processing and machine learning methods that are concerned with finding abstract and meaningful representations of data for inference tasks, while being computationally efficient and robust to deviations from assumptions. Technological advances in data gathering systems, as well as emergence of powerful and inexpensive processors, have led to an everincreasing need for new machine learning and signal processing techniques that not only can extract information from the data, but also are able to compress/summarize it, detect outliers, and to create meaningful representations of data in a low-dimensional space. The recent success of machine learning algorithms can be arguably attributed to the new methods to find better data representations, also known as *features*. It is safe to say that most of the efforts in designing new machine learning or signal processing methods go into finding effective application-specific data representations, either by using domain-specific knowledge or by consuming huge amounts of data for training. This work presents new contributions on three important and closely related tasks: (i) supervised representation learning, (ii) unsupervised representation learning, and (iii) fault tolerant data analysis. In the following, each of these tasks are discussed in detail and the contributions of this doctoral thesis are outlined. A more detailed discussion of the prior work and the specific contributions of each chapter is provided in the corresponding chapter text.

1.1 Supervised Representation Learning

The goal of representation learning is to find a useful, and oftentimes low dimensional, representation of data that makes the task at hand easier. For instance, in the classification task, if we can transform the data into a space such that different classes are far from each other, the classification

task becomes trivial. *Supervised* representation learning can be used whenever enough labelled data is available for training and the desired transformation can be found by using the input-output pairs to optimize a well-defined cost function, e.g., classification accuracy. The recent empirical success of machine learning is evidently owed to the rediscovery of neural networks, and in particular deep neural networks (DNNs). Deep neural networks have progressed rapidly during the last few years, achieving outstanding, sometimes super human, performance [9]. DNNs can be described as composition of many nonlinear *feature extractors*, making them very efficient in learning meaningful representation of data and in learning complex mappings of data. Each of the feature extractors consists of a linear and a nonlinear transformation.

For the task of supervised representation learning, we investigate different aspects of DNNs. Specifically, we discuss how some DNNs can become very deep (Chapter 3), how the embedding space generated by them can become robust to outliers (Chapter 4), and how to use such embedding spaces to retrieve a subset of samples in the context of summarization (Chapter 5) and query-based search (Chapter 6).

It is known that the depth of the network, i.e., number of stacked feature extractors, is of decisive significance. It is shown that as the networks become deeper, they are capable of representing more complex mappings [10]. However, deeper networks are notoriously harder to train. As the number of layers is increased, optimization issues arise and, in particular, avoiding vanishing/exploding gradients is essential to optimization stability of such networks. Augmenting neural networks with skip connections, as introduced in the so-called ResNet architecture[11, 12], surprised the community by enabling the training of networks of more than 1,000 layers with significant performance gains.

In Chapter 3, we decipher ResNet by analyzing the effect of skip connections, and put forward new theoretical results on the advantages of identity skip connections in neural networks. We prove

that the skip connections in the feature extractors facilitate preserving the norm of the gradient and lead to stable back-propagation, which is desirable from optimization perspective. We also show that as more feature extractors are stacked, the norm-preservation of the network is enhanced. Furthermore, we propose an efficient method to regularize the singular values of the convolution operator to make the ResNet extra norm-preserving. Our numerical investigations demonstrate that the learning dynamics and the classification performance of ResNet can be improved by making it even more norm preserving.

In Chapter 4, we show how we can manipulate DNNs, and in particular ResNets, to generate data representations that are robust to outliers and anomalies. Many conventional machine learning methods are being designed and deployed under *closed-set* assumptions, meaning that training data contains samples from all the possible classes that the classifier will encounter during testing. Of course, such assumption does not hold in many applications; as it may not be possible to cover every potential input class in the training set. Thus, the goal of open-set classifiers is to detect out-of-distribution (OOD) samples; the input instances that do not belong to any of the training classes. In general, OOD detection techniques try to either use the class membership probabilities as a measure of uncertainty [13–16], or define a measure of similarity between the input samples and the training dataset in a feature space [17–19]. In Chapter 4, we argue that OOD samples can be detected far more easily if the training data is embedded into a low-dimensional space, such that the embedded training samples (or features) lie on a union of 1-dimensional subspaces.

We show that such embedding of the in-distribution (ID) samples provides us with two main advantages. First, due to compact representation in the feature space, OOD samples are less likely to occupy the same region as the known classes. Second, the first singular vector of samples belonging to a 1-dimensional subspace is their robust representative. Motivated by these findings, we train a deep neural network such that the ID samples are embedded onto a union of 1-dimensional subspaces. At the test time, employing Monte Carlo sampling, input samples are detected as OOD

if they occupy the region corresponding to the ID samples with probability 0. Spectral components of the ID samples are used as robust representative of this region.

Chapter 5 investigates how we can use the representations generated by neural networks to effectively summarize huge dataset, by selecting a few *representatives*. As mentioned earlier, deep learning based systems employ very large numbers of inputs. However, processing, labeling, and communication of a large number of input data have remained challenging. Therefore, novel machine learning methods that make the best use of a significantly less amount of data are of great interest. For example, active learning (AL) [20] aims at addressing this problem by training a model using a small number of labeled data, testing the trained model on a large number of unlabelled data, and then querying the labels of some selected data, which then are used for training a new model. In this context, preserving the underlying structure of data by a succinct format is an essential concern. Chapter 5 presents a fast and accurate data selection method, in which the selected samples are optimized to span the subspace of all data. We show how our efficient algorithm (linear complexity w.r.t. the number of data), in conjunction with deep feature extractors can achieve superior performance in different application such as active learning for video action recognition; learning using representatives; and video summarization.

Similarly, Chapter 6 discusses a framework that uses the embedding space generated by neural networks to search and retrieve images from huge datasets. Specifically, we introduce a new face image retrieval task, where the input face query is augmented by both a modification vector that specifies the desired adjustments to the facial attributes and a preference vector that assigns different levels of importance to different attributes. For example, a user can ask for retrieving images similar a query image, but with a different hair color and no preference for absence/presence of eyeglasses in the results. To achieve this, we propose to learn a set of disentangled basis vectors in the latent space of Generative Adversarial Networks (GANs) [21]. We show how such basis vectors can be employed to adjust the attributes, to define an attribute-weighted distance metric, and

to retrieve similar face images. To disentangle various semantics, we propose to enforce orthogonality and sparsity constraints on the basis vectors corresponding to the attributes. We show how these constraints lead to more precise and easier control of attributes and better image retrieval.

1.2 Unsupervised Representation Learning

In many applications the input-output pairs are not available and the representation learning task need to be carried out in an unsupervised setting. In such cases, prior knowledge on the signal of interest can be manipulated to reveal the hidden low-dimensional representations. Such representations are preferred to be a succinct summary of the original raw data and are usually employed for denoising, outlier rejection, missing data estimation, compression, and/or revealing the latent structures. For the task of unsupervised representation learning, we introduce two new methods, namely adaptive non-uniform compressive sensing (Chapter 7) and nonnegative matrix factorization with piece-wise constant priors (Chapter 8). We will show how they can be used for compression and missing data estimation.

In Chapter 7, adaptive non-uniform compressive sampling (ANCS) of time-varying signals, which are sparse in a proper basis, is introduced. Compressed sensing (CS) [22, 23] states that most of the signals of scientific interest can be approximated very accurately using a smaller number of measurements, compared to the dimension of the signal. For that, the signal needs to be sparse or have a sparse representation in terms of proper sparsifying bases. This observation has a huge impact in signal processing, machine learning, and statistics. Chapter 7 considers the problem of reconstructing a correlated time series of such compressible vectors from their noisy undersampled measurement. In many applications, coefficients of the signal of interest have different importance levels and the *region of interest* (ROI) is not known *a priori*. For instance, the salient area in a sequence of video frames or support of a sparse signal can be considered as the ROI. ANCS

employs the measurements of previous time steps to design the measurement matrix and distribute the sensing energy among the coefficients more intelligently. To this aim, a Bayesian inference method is proposed, which introduces an unsupervised learning component to the traditional CS framework and improves the reconstruction ability in the ROI. ANCS has been shown to be an effective method in designing sampling hardware [24].

In Chapter 8, we will discuss how sparsity constraint can be exploited to facilitate imposing structures on the latent representation of the signal. We show the effectiveness of the method in estimating the missing entries in data collected by a network of spectrum sensors. Particularly, we propose a missing spectrum data recovery technique using Nonnegative Matrix Factorization (NMF). It is shown that the spectrum measurements collected from sensors can be factorized as product of a channel gain matrix times an activation matrix. Then, an NMF method with piece-wise constant activation coefficients is introduced to analyze the measurements and estimate the missing spectrum data. However, solving the factorization problem with piece-wise continuity constraint is not an easy task. Thus, a Majorization-Minimization technique is developed to solve the proposed optimization problem. The proposed technique is able to accurately and efficiently estimate the missing spectrum data in the presence of noise and fading.

1.3 Fault Tolerant Data Analysis

Another desirable feature for machine learning and signal processing frameworks is the ability to extract information from the raw data and/or the representations of the data in a robust, fault tolerant, manner. Due to the proliferation of inexpensive hardware for data gathering and processing units, the data is now being gathered and processed by many, possibly unreliable, devices. Thus, it is necessary for any data analysis framework to be able to detect and handle faulty data, and even faulty processing units. In this thrust, we discuss two possible approaches on how we can robustify

the data processing framework to faults in both the collection (Chapter 10) and processing (Chapter 9) phases.

Particularly, in Chapter 9, we will introduce an *asynchronous* parallel algorithm to solve a sparse recovery problem. In parallel algorithms, the task is partitioned among many processing units to reduce the computational and storage requirements, and/or to preserve the privacy. Asynchronous methods are highly desirable, as some subset of the processing units does not need to wait for another subset to finish their tasks, unlike synchronous techniques. This makes the asynchronous parallel algorithms robust to slow and non-functioning nodes. However, asynchronous parallel algorithms are often studied for separable optimization problems where the component objective functions are *sparse*, or act on only a few components of the unknown variable. One challenge to developing asynchronous approaches for sparse recovery is that the optimization formulation of this problem has dense component objective functions. However, the assumed sparsity of the signal may be exploited in an asynchronous parallel approach. In Chapter 9, we propose such an approach where multiple processors asynchronously infer hidden variables that estimate the support of the signal in a Bayesian manner.

Finally, in Chapter 10, we will discuss how a particular class of optimization problems known as generalized trust region subproblems (GTRS) can be made robust against faulty measurements. Particularly, the problem of target localization in the presence of outlying sensors is tackled. This problem is important in practice because in many real-world applications the sensors might report irrelevant data unintentionally or maliciously. We propose a localization method based on robust statistics, seeking to eliminate the effect of outliers. The problem is formulated by applying robust statistics techniques on squared range measurements and two different approaches to solve the problem are proposed. The first approach is computationally efficient; however, only the objective convergence is guaranteed theoretically. On the other hand, the whole-sequence convergence of the second approach is established. To enjoy the benefit of both approaches, they are integrated

to develop a hybrid algorithm that offers computational efficiency and theoretical guarantees. The algorithms are evaluated for different simulated and real-world scenarios. The numerical results show that the proposed methods meet the Cràmer-Rao lower bound (CRLB) for a sufficiently large number of measurements. When the number of the measurements is small, the proposed position estimator does not achieve CRLB though it still outperforms several existing localization methods.

CHAPTER 2: BACKGROUND

In this chapter, we discuss some of the background needed to facilitate the understanding of the contents of this dissertation. Specifically, in Chapter 3, Chapter 4, Chapter 5, and Chapter 6, we study or modify the inner workings of *deep neural networks* and in Chapter 7 and Chapter 9 we propose techniques to improve *compressive sensing* systems. Thus, it is worthwhile to go over a brief summary of these machine learning and signal processing paradigms.

2.1 Deep Neural Networks

Deep neural networks can be considered as the generalization of classical classification methods, such as support vector machines (SVMs). Their success during the last decade is largely owed to the development of new optimization techniques, collection of large-scale datasets, and advancements in processing hardware. A simple binary classification problem can be defined as follows: Given a set of vectors $\{\mathbf{x}_n\}_{n=1}^N$, their corresponding binary labels $\{y_n\}_{n=1}^N$, and a new unlabelled vector \mathbf{x} , we want to be able to guess its label \hat{y} . To solve this problem, an SVM model can be trained by optimizing the following objective function:

$$L = \frac{1}{n} \sum_{n=1}^N \max(0, 1 - y_n(\mathbf{w}^T \mathbf{x}_n - b)) + \lambda \|\mathbf{w}\|_2^2,$$

where \mathbf{w} and b are the learnable model parameters. The first term is the classification loss, namely hinge loss, and the second term is ℓ_2 -regularization term on the weights. At the test time, we can estimate the label as $\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} - b)$, where $\sigma(\cdot)$ is the step function or its differentiable counterpart, the sigmoid function. In words, if the inner product of \mathbf{x} with \mathbf{w} is larger than b , we will assign label 1 to \mathbf{x} , and 0 otherwise. In this setting, the direction of vector \mathbf{w} is perpendicular to

the decision boundary and b determines the position of the decision boundary along that direction. Thus, \mathbf{w} and b characterize the decision boundary of our classifier. Figure 2.1 illustrates this for a simple example in a 2-dimensional space.

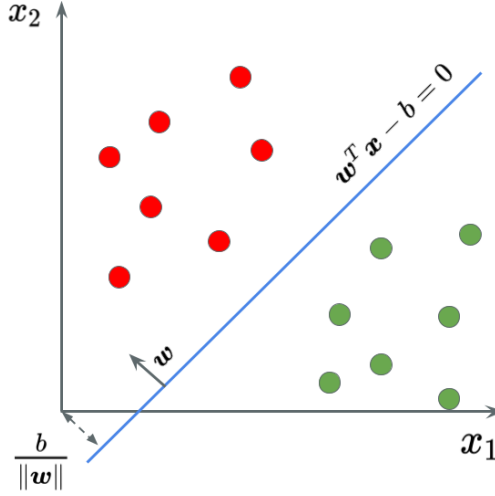


Figure 2.1: An illustration of the decision boundary learned by an SVM classifier for a binary classification problem in a 2-dimensional space. The vector \mathbf{w} is perpendicular to the decision boundary and the scalar b determines its position with respect to the origin.

This formulation can be generalized to multiple classes, by finding a decision boundary for each class, $\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{x} - \mathbf{b})$, where each row of \mathbf{W} is a vector perpendicular to the decision boundary of its corresponding class, each element in \mathbf{b} determines the position of its corresponding decision boundary, and $\sigma(\cdot)$ is an elementwise nonlinear function. The classification function $\sigma(\mathbf{W}\mathbf{x} - \mathbf{b})$, which is a composition of a linear operator and a nonlinear operator, divides the input space into multiple regions, characterized by the learned decision boundaries. We can create exponentially more regions if we stack several of these functions on top of each other. This is the idea behind Multilayer Perceptron (MLP). MLP is one of the earliest versions of neural network and consists of only a few hidden layers. Each layer has a linear operation (matrix multiplication), followed by an element-wise nonlinear operator (activation function), similar to our SVM example $\sigma(\mathbf{W}\mathbf{x} - \mathbf{b})$.

The next development in the neural network literature was the introduction of the convolutional neural networks. In many applications, such as image, video, and audio processing, the signal of interest is shift-invariant. Thus, by employing a shift-invariant linear operator, i.e. convolution, we can hard-code this shift invariance assumption into our model. This leads to significant reduction in the number of parameters, easier implementation, and reduction in the size of the solution space. As mentioned earlier, deeper neural networks are able to separate their input space into exponentially more linear response regions than their shallow counterparts, despite using the same number of computational units [10]. In other words, a shallow network requires exponentially many more hidden units than a deep network. Thus, in general, deeper neural networks are desirable, as they lead to parameter and data efficiency.

To train such models, similar to the SVM example, the parameters of the model can be optimized by minimizing some loss function using gradient descent. However, in the case of neural networks with multiple layers we need to *backpropagate* the gradient through the layers using the chain rule. Specifically, since the output of the l^{th} layer can be written as $\mathbf{x}_l = F_l(\mathbf{x}_{l-1}) = \sigma(\mathbf{W}_l \mathbf{x}_{l-1} - \mathbf{b}_l)$, the gradient of the loss L with respect to \mathbf{x}_{l-1} can be calculated as

$$\frac{\partial L}{\partial \mathbf{x}_{l-1}} = \frac{\partial \mathbf{x}_l}{\partial \mathbf{x}_{l-1}} \frac{\partial L}{\partial \mathbf{x}_l} = \mathbf{W}_l^T \sigma'(\mathbf{W}_l \mathbf{x}_{l-1} - \mathbf{b}_l) \frac{\partial L}{\partial \mathbf{x}_l}.$$

Thus, given the gradient at l^{th} layer, $\frac{\partial L}{\partial \mathbf{x}_l}$, we can calculate the the gradient at the $(l-1)^{\text{th}}$ layer $\frac{\partial L}{\partial \mathbf{x}_{l-1}}$, using a matrix multiplication. However, in deeper neural networks, due to the multiplication effect, the gradient can increase/decrease exponentially with l . This leads to stability issues and/or complete of halt of the training. This phenomena, which is referred to as exploding/vanishing gradient, makes the training of the deep neural networks very difficult. In other words, although increasing the number of layers increases the representational ability of the model, it hurts the performance, due to the optimization issues.

This degradation problem was addressed by the *deep residual learning* framework [12]. In this framework, each block in the model tries to fit a residual mapping, i.e., $\mathbf{x}_l = \mathbf{x}_{l-1} + F_l(\mathbf{x}_{l-1})$, instead of the mapping itself $\mathbf{x}_l = F_l(\mathbf{x}_{l-1})$. This means that the propagation of the gradients through the layers will have an additive form, instead of a multiplicative form:

$$\frac{\partial L}{\partial \mathbf{x}_{l-1}} = \left(1 + \frac{\partial F_l(\mathbf{x}_{l-1})}{\partial \mathbf{x}_{l-1}}\right) \frac{\partial L}{\partial \mathbf{x}_l} = \frac{\partial L}{\partial \mathbf{x}_l} + \frac{\partial F_l(\mathbf{x}_{l-1})}{\partial \mathbf{x}_{l-1}} \frac{\partial L}{\partial \mathbf{x}_l}.$$

This modification in the model enabled the deep residual networks to be trained without difficulties and lowered both the training and generalization error of neural networks. In this thesis, we will discuss the learning dynamics of residual networks (ResNets) in more details in Chapter 3.

Neural networks have also been used to generate synthetic, but realistic, samples. To achieve this, an adversarial training framework was proposed by Goodfellow et. al. [21]. In this scheme, two neural networks, namely the generator and the discriminator, are trained simultaneously. The discriminator is trained such that it can distinguish between real and fake images, while the generator is trained to be able to generate realistic fake images and to fool the discriminator. During the training, the discriminator network becomes better and better at detecting fake synthetic images, while the generator network becomes better at fooling the discriminator, leading to more realistic fake images. Such architecture, known as generative adversarial network (GAN), has proved to be able to generate hyper-realistic images. In Chapter 6, we use the latent space created by a GAN to devise an image retrieval framework.

2.2 Compressive Sensing

Many real-world signals, including images, videos, wideband radio signals, and biomedical signals, are sparse or can be sparsely represented in some proper basis, e.g. Fourier or wavelet domain.

This means that the signal of interest can be written as a linear combination of only a few of the basis vectors/functions in the basis set. Compressive sensing [22, 25] enables us to recover such compressible signals from their undersampled random projections. This means that we can potentially sample the signal at rates much lower than the Nyquist rate, while not losing much in terms of reconstruction accuracy. This is specially important in cases where sampling the signal at Nyquist rate is very expensive, such as wideband signals or infrared imaging.

Specifically, a vector $\mathbf{x} \in \mathbb{R}^N$ is considered sparse in some basis if it can be represented as a linear combination of only $k \ll N$ of the basis vectors. According to the compressive sensing paradigm, we can recover \mathbf{x} from only $M \ll N$ random measurements. Such measurements are usually obtained as $\mathbf{y} = \Phi \mathbf{x}$, where $\Phi \in \mathbb{R}^{M \times N}$ is a well-chosen random matrix. This means that each entry in the measurement vector \mathbf{y} is a random linear combination of the entries in the signal of interest \mathbf{x} . The recovery of the original signal involves solving an under-determined system of equations, with possibly infinite solutions. But our prior knowledge of sparsity of \mathbf{x} enables us to find a unique solution. It has been shown that under certain conditions, we can recover \mathbf{x} by solving the following optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \Phi \mathbf{x}$$

where $\|\cdot\|_1$ is the ℓ_1 norm operator, i.e., the sum of the absolute values of the vector. This optimization problem is referred to as basis pursuit (BP) [22]. The condition under which BP can find the solution with very high probability is referred to as restricted isometry property (RIP). Specifically, if for any $2k$ -sparse vector \mathbf{v} , the measurement matrix satisfies:

$$(1 - \delta_{2k})\|\mathbf{v}\|_2^2 \leq \|\Phi \mathbf{v}\|_2^2 \leq (1 + \delta_{2k})\|\mathbf{v}\|_2^2,$$

for some $0 < \delta_{2k} < 1$, BP can recover k -sparse signals using Φ with very high probability.

Intuitively, RIP states that the measurement matrix does not change the norm of any $2k$ -sparse vector *much*. This is important because if we have two k -sparse signals \mathbf{x}_1 and \mathbf{x}_2 , the difference vector $\mathbf{x}_1 - \mathbf{x}_2$ can be a $2k$ -sparse vector. Thus, in order to be able to distinguish between \mathbf{x}_1 and \mathbf{x}_2 , Φ needs to preserve the distance between them, i.e., $\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$, as much as possible (smaller the δ_{2k} , the better). RIP is obeyed by many types of matrices such as Gaussian random matrices. In Chapter 7 and Chapter 9, we show how we can use Bayesian inference in measurement or reconstruction steps to improve the performance.

CHAPTER 3: NORM-PRESERVATION: WHY RESIDUAL NETWORKS CAN BECOME EXTREMELY DEEP?

It is known that the depth of the network, i.e., number of stacked layers, is of decisive significance. It is shown that as the networks become deeper, they are capable of representing more complex mappings [10]. However, deeper networks are notoriously harder to train. As the number of layers is increased, optimization issues arise and, in particular, avoiding vanishing/exploding gradients is essential to optimization stability of such networks. Batch normalization, regularization, and initialization techniques have shown to be useful remedies for this problem [26, 27]¹.

Furthermore, it has been observed that as the networks become increasingly deep, the performance gets saturated or even deteriorates [11]. This problem has been addressed by many recent network designs [11, 12, 29, 30]. All of these approaches use the same design principle: skip connections. This simple trick makes the information flow across the layers easier, by bypassing the activations from one layer to the next using skip connections. Highway Networks [29], ResNets [11, 12], and DenseNets [30] have consistently achieved state-of-the-art performances by using skip connections in different network topologies. The main goal of skip connection is to enable the information to flow through many layers without attenuation. In all of these efforts, it is observed empirically that it is crucial to keep the information path *clean* by using identity mapping in the skip connection. It is also observed that more complicated transformations in the skip connection lead to more difficulty in optimization, even though such transformations have more representational capabilities [12]. This observation implies that *identity* skip connection, while provides adequate representational ability, has a great feature of optimization stability, enabling deeper well-behaved

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, N. Rahnavard, and M. Shah, “Norm-Preservation: Why Residual Networks Can Become Extremely Deep?,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020, © 2020 IEEE [28].

networks.

Since the introduction of Residual Networks (ResNets) [11, 12], there have been some efforts on understanding how the residual blocks may help the optimization process and how they improve the representational ability of the networks. Authors in [31] showed that skip connection eliminates the singularities caused by the model non-identifiability. This makes the optimization of deeper networks feasible and faster. Similarly, to understand the optimization landscape of ResNets, authors in [32] prove that linear residual networks have no critical points other than the global minimum. This is in contrast to plain linear networks, in which other critical points may exist [33]. Furthermore, authors in [34] show that as depth increases, gradients of plain networks resemble white noise and become less correlated. This phenomenon, which is referred to as *shattered gradient* problem, makes training more difficult. Then, it is demonstrated that residual networks reduce shattering, compared to plain networks, leading to numerical stability and easier optimization.

In this chapter, we present and analytically study another desirable effect of identity skip connection: *the norm preservation of error gradient*, as it propagates in the backward path. We show theoretically and empirically that each residual block in ResNets is *increasingly norm-preserving*, as the network becomes *deeper*. This interesting result is in contrast to hypothesis provided in [35], which states that residual networks avoid vanishing gradient *solely* by shortening the effective path of the gradient.

Furthermore, we show that identity skip connection enforces the norm-preservation during the training, leading to well-conditioning and easier training. This is in contrast to the initialization techniques, in which the initialization distribution is modified to make the training easier [26, 36]. This is done by keeping the variance of weights gradient the same across layers. However, as observed in [36] and verified by our experiments, using such initialization methods, although the network is initially fairly norm-preserving, the norms of the gradients diverge as training pro-

gresses.

We analyze the role of identity mapping as skip connection in the ResNet architecture from a theoretical perspective. Moreover, we use the insight gained from our theoretical analysis to propose modifications to some of the building blocks of the ResNet architecture. Two main contributions of this chapter are as follows.

- **Proof of the Norm Preservation of ResNets:** We show that having identity mapping in the shortcut path leads to norm-preserving building blocks. Specifically, identity mapping shifts all the singular values of the transformations towards 1. This makes the optimization of the network much easier by preserving the magnitude of the gradient across the layers. Furthermore, we show that, perhaps surprisingly, *as the network becomes deeper, its building blocks become more norm-preserving*. Hence, the gradients can flow smoothly through very deep networks, making it possible to train such networks. Our experiments validate our theoretical findings.
- **Enhancing Norm Preservation:** Using insights from our theoretical investigation, we propose important modifications to the *transition blocks* in the ResNet architecture. The transition blocks are used to change the number of channels and feature map size of the activations. Since these blocks do not use identity mapping as the skip connection, in general, they do not preserve the norm of the gradient. We propose to change the dimension of the activations in a norm preserving manner, such that the network becomes even more norm-preserving. For that, we propose a computationally efficient method to set the nonzero singular values of the convolution operator, without using singular value decomposition. We refer to the proposed architecture as Procrustes ResNet (ProcResNet). Our experiments demonstrate that the proposed norm-preserving blocks are able to improve the optimization stability and performance of ResNets.

The rest of the chapter is organized as follows. In Section 3.1, the theoretical results and the bounds for norm-preservation of linear and nonlinear residual networks are presented. Then, in Section 3.2, we show how to enhance the norm preservation of the residual networks by introducing a new computationally efficient regularization of convolutions. To verify our theoretical investigation and to demonstrate the effectiveness of the proposed regularization, we provide our experiments in Section 3.3. Finally, Section 3.4 draws conclusions.

3.1 Norm-Preservation of Residual Networks

Our following main theorem states that, under certain conditions, a deep residual network representing a nonlinear mapping is norm-preserving in the backward path. We show that, at each residual block, the norm of the gradient with respect to the input is close to the norm of gradient with respect to the output. In other words, *the residual block with identity mapping, as the skip connection, preserves the norm of the gradient in the backward path*. This results in several useful characteristics such as avoiding vanishing/exploding gradient, stable optimization, and performance gain.

Suppose we want to represent a nonlinear mapping $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with a sequence of L non-linear residual blocks of form:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + F_l(\mathbf{x}_l). \quad (3.1)$$

As illustrated in Figure 3.1(b), \mathbf{x}_l and \mathbf{x}_{l+1} represent respectively the input and output at l^{th} layer. $F_l(\mathbf{x}_l)$ is the residual transformation learned by the l^{th} layer. Before presenting the theorem, we lay out the following assumptions on \mathcal{F} .

Assumption 3.1. *The function $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is differentiable, invertible, and satisfies the following conditions:*

- (i) $\forall \mathbf{x}, \mathbf{y}, \mathbf{z}$ with bounded norm, $\exists \alpha > 0$, $\|(\mathcal{F}'(\mathbf{x}) - \mathcal{F}'(\mathbf{y}))\mathbf{z}\| \leq \alpha \|\mathbf{x} - \mathbf{y}\| \|\mathbf{z}\|$,
- (ii) $\forall \mathbf{x}, \mathbf{y}$ with bounded norm, $\exists \beta > 0$, $\|\mathcal{F}^{-1}(\mathbf{x}) - \mathcal{F}^{-1}(\mathbf{y})\| \leq \beta \|\mathbf{x} - \mathbf{y}\|$, and
- (iii) $\exists \mathbf{x}$ with bounded norm such that $\text{Det}(\mathcal{F}'(\mathbf{x})) > 0$,

α and β are constants, independent of network size and architecture. Also, we assume that the domain of inputs is bounded. By rescaling inputs, we can assume, without loss of generality, that $\|\mathbf{x}_1\|_2 \leq 1$ for any input \mathbf{x}_1 .

We would like to emphasize the point that these assumptions are on the mapping that we are trying to represent by the network, not the network itself. Thus, assumptions are independent of architecture. Assumptions (i) and (ii) mean that the function \mathcal{F} is smooth, Lipschitz continuous, and its inverse is differentiable. The practical relevance of invertibility assumption is justified by the success of reversible networks [37–39]. Reversible architectures look for the true mapping \mathcal{F} *only* in the space of invertible functions and it is shown that imposing such strict assumption on the architecture does not hurt its representation ability [38]. Thus, the mapping \mathcal{F} is either invertible or can be well approximated by an invertible function, in many scenarios. However, unlike the reversible architectures, we do not assume residual blocks or the residual transformations, $F_l(\cdot)$, are invertible, which makes the assumption less strict. Furthermore, our extensive experiments in Section 3.3 show that our theoretical analysis, which is based on these assumptions, hold. This is further empirical justification that these assumptions are relevant in practice. Finally, Assumption (iii) is without loss of generality [32, 40].

Theorem 3.1. *Suppose we want to represent a nonlinear mapping $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$, satisfying Assumption 3.1, with a sequence of L non-linear residual blocks of form $\mathbf{x}_{l+1} = \mathbf{x}_l + F_l(\mathbf{x}_l)$. There exists a solution such that for all residual blocks we have:*

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2,$$

where $\delta = c \frac{\log(2L)}{L}$, $\mathcal{E}(\cdot)$ is the cost function, and $c = c_1 \max\{\alpha\beta(1 + \beta), \beta(2 + \alpha) + \alpha\}$ for some $c_1 > 0$. α and β are constants defined in Assumption 3.1.

Proof. See Appendix A.1. □

This theorem shows that the mapping \mathcal{F} can be represented by a sequence of L non-linear residual blocks, such that the norm of the gradient does not change significantly, as it is backpropagated through the layers. *One interesting implication of Theorem 3.1 is that as L , the number of layers, increases, δ becomes smaller and the solution becomes more norm-preserving.* This is a very desirable feature because vanishing or exploding gradient often occurs in deeper network architectures. However, by utilizing residual blocks, as more blocks are stacked, the solution becomes extra norm-preserving.

Now that we proved such a solution exists, we show why residual networks can remain norm preserving throughout the training. For that, we consider the case where $F_l(\mathbf{x}_l)$ consists of two layers of convolution and nonlinearity. The following corollary shows the bound on norm preservation of the residual block depends on the norm of the weights. Therefore, if we bound the optimizer to search only in the space of functions with small norms, we can ensure that the network will remain norm preserving throughout the training. Therefore, any critical point in this space is also norm-preserving. On the other hand, based on Theorem 3.1, we know that at least one norm preserving solution exists. It is also known that, under certain conditions, any critical point achieved during optimization of ResNets is a global minimizer, meaning that it achieves the same loss function value as the global minimum[32, 40, 41]. Thus, this result implies that ResNets are able to maintain norm-preservation throughout the training and if they converge, the solution is a norm-preserving global minimizer. The conclusions of the corollary can be easily generalized for residual block with more than two layers.

Corollary 3.1. *Suppose a network contains non-linear residual blocks of form $\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{W}_l^{(2)} \rho(\mathbf{W}_l^{(1)} \rho(\mathbf{x}_l))$, where $\rho(\cdot)$ is an element-wise non-linear operator with bounded derivative, i.e., $0 \leq \frac{\partial \rho_n(\mathbf{x}_l)}{\partial x_{l,n}} \leq c_\rho, \forall n = 1, \dots, N$. Then, we have:*

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2$$

$$\text{and } \delta = c_\rho^2 \|\mathbf{W}_l^{(1)}\|_2 \|\mathbf{W}_l^{(2)}\|_2.$$

Proof. See Appendix A.3 □

Here, $\|\cdot\|_2$ is the induced matrix norm and is the largest singular value of the matrix, which is known to be upper bounded by the entry-wise ℓ_2 norm. This means that norm-preservation is enforced throughout the training process, as long as the norm of the weights are small, not just at the beginning of the training by good initialization. This is the case in practice, since the weights of the network are regularized either explicitly using ℓ_2 regularization, also known as weight decay, or implicitly by the optimization algorithm [42, 43]. Thus, the gradients will have very similar magnitudes at different layers, and this leads to well-conditioning and faster convergence [36].

Although Theorem 3.1 holds for linear blocks as well, we can derive tighter bounds for linear residual blocks by taking a slightly different approach. For that, we model each linear residual block as:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{W}_l \mathbf{x}_l, \tag{3.2}$$

where, $\mathbf{x}_l, \mathbf{x}_{l+1} \in \mathbb{R}^N$ are respectively the input and output of the l^{th} residual block, with dimension N . The weight matrix $\mathbf{W}_l \in \mathbb{R}^{N \times N}$ is the tunable linear transformation. The goal of learning is to compute a function $\mathbf{y} = \mathcal{M}(\mathbf{x}, \mathcal{W})$, where $\mathbf{x} = \mathbf{x}_1$ is the input, $\mathbf{y} = \mathbf{x}_{L+1}$ is its corresponding output, and \mathcal{W} is the collection of all adjustable linear transformations, i.e., $\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_L$. In

the case of simplified linear residual networks, function $\mathcal{M}(\mathbf{x}, \mathcal{W})$ is a stack of L residual blocks, as formulated in (3.2). Mathematically speaking, we have:

$$\mathbf{y} = \mathcal{M}(\mathbf{x}, \mathcal{W}) = \prod_{l=1}^L (\mathbf{I} + \mathbf{W}_l) \mathbf{x}, \quad (3.3)$$

where \mathbf{I} is an $N \times N$ identity matrix. $\mathcal{M}(\mathbf{x}, \mathcal{W})$ is used to learn a linear mapping $\mathbf{R} \in \mathbb{R}^{N \times N}$ from its inputs and outputs. Furthermore, assume that \mathbf{y} is contaminated with independent identically distributed (i.i.d) Gaussian noise, i.e., $\hat{\mathbf{y}} = \mathbf{R}\mathbf{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a zero mean noise vector with covariance matrix \mathbf{I} . Hence, our objective is to minimize the expected error of the maximum likelihood estimator as:

$$\min_{\mathcal{W}} \mathcal{E}(\mathcal{W}) = \mathbb{E} \left\{ \frac{1}{2} \|\hat{\mathbf{y}} - \mathcal{M}(\mathbf{x}, \mathcal{W})\|_2^2 \right\}, \quad (3.4)$$

where the expectation \mathbb{E} is with respect to the population (\mathbf{x}, \mathbf{y}) . The following theorem states the bound on the norm preservation of the linear residual blocks.

Theorem 3.2. *For learning a linear map, $\mathbf{R} \in \mathbb{R}^{N \times N}$, between its input \mathbf{x} and output \mathbf{y} contaminated with i.i.d Gaussian noise, using a network consisting of L linear residual blocks of form $\mathbf{x}_{l+1} = \mathbf{x}_l + \mathbf{W}_l \mathbf{x}_l$, there exists a global optimum for $\mathcal{E}(\cdot)$, as defined in (3.4), such that for all residual blocks we have*

$$(1 - \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} \right\|_2 \leq (1 + \delta) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2$$

for $L \geq 3\gamma$, where $\delta = \frac{c}{L}$, $c = 2(\sqrt{\pi} + \sqrt{3\gamma})^2$, and $\gamma = \max(|\log \sigma_{\max}(\mathbf{R})|, |\log \sigma_{\min}(\mathbf{R})|)$, where $\sigma_{\max}(\mathbf{R})$ and $\sigma_{\min}(\mathbf{R})$, respectively, are maximum and minimum singular values of \mathbf{R} .

Proof. See Appendix A.2 □

Similar to the nonlinear residual blocks, the linear blocks become more norm-preserving as we

increase the depth. However, the linear blocks become norm-preserving at a faster rate. The gradient norm ratio for the linear blocks approaches 1 with a rate of $\mathcal{O}(\frac{1}{L})$, while this ratio for nonlinear blocks approaches 1 with a rate of $\mathcal{O}(\frac{\log(L)}{L})$.

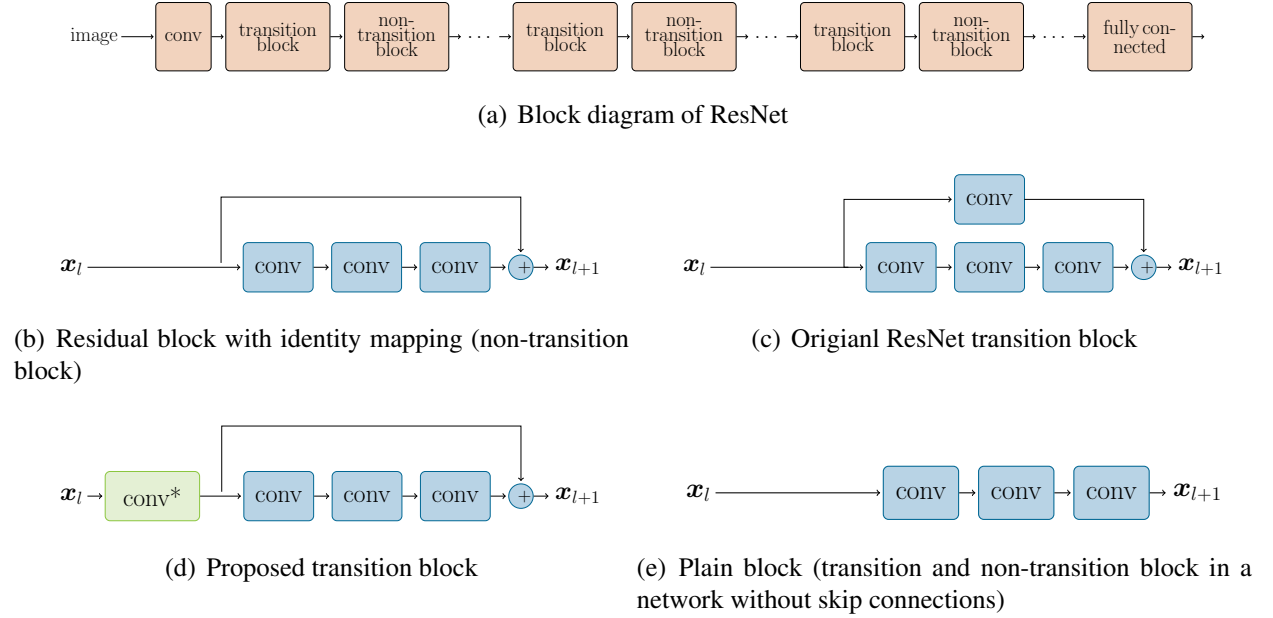


Figure 3.1: ResNet architecture and its building blocks. Each conv block represents a sequence of batch normalization, ReLU, and convolution layers. conv* block represents the regularized convolution layer.

3.2 Procrustes Residual Network

As depicted in Figure 3.1(a), residual networks contain four different types of blocks: (i) convolution layer (first layer), (ii) fully connected layer (last layer), (iii) transition blocks (which change the dimension) as depicted in Figure 3.1(c), and (iv) residual blocks with identity skip connection, as illustrated in Figure 3.1(b), which we also refer to as non-transition blocks. Theoretical investigation presented in Section 3.1 holds only for residual blocks with identity mapping as the skip connection. Such identity skip connection cannot be used in the transition blocks, since the

size of the input is not the same as the size of output. If the benefits of residual networks can be explained, at least partly, by norm-preservation, then one can improve them by alternative methods for preserving the norm. In this section, we propose to modify the transition blocks of ResNet architecture, to make them norm-preserving. Due to multiplicative effect through the layers, making these layers norm-preserving may be important, although they make up a small portion of the network. In the following, we discuss how to preserve the norm of the back-propagated gradients across all the blocks of the network.

As depicted in Figure 3.1(c), in the original ResNet architecture, the dimension changing blocks, also known as transition blocks, use 1×1 convolution with stride of 2 in their skip connections to match the dimension of input and output activations. Such transition blocks are not norm-preserving in general.

Figure 3.1(d) shows the block diagram of the proposed norm-preserving transition block. To change the dimension in a norm-preserving manner, we utilize a norm preserving convolution layer, conv^* . For that, we project the convolution kernel onto the set of norm preserving kernels by setting its singular values. Here, we show how we can make the convolution layer norm preserving by regularizing the singular values, without using singular value decomposition. Specifically, the gradient of a convolution layer with kernel of size k , with c input channels, and d output channels can be formulated as:

$$\Delta_x = \hat{\mathbf{W}} \Delta_y, \quad (3.5)$$

where Δ_x and Δ_y respectively are the gradients with respect to the input and output of the convolution. Δ_y is an $n^2 d$ dimensional vector, representing $n \times n$ pixels in d output channels, and Δ_x is an $n^2 c$ dimensional vector, representing the gradient at the input. Furthermore, $\hat{\mathbf{W}}$ is an $n^2 c \times n^2 d$ dimensional matrix embedding the back-propagation operation for the convolution layer. We can

represent this linear transformation as:

$$\Delta_{\mathbf{x}} = \sum_{j=1}^{n^2 c} \sigma_j \mathbf{u}_j \langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle, \quad (3.6)$$

where $\{\sigma_j, \mathbf{u}_j, \mathbf{v}_j\}$ is the set of singular values and singular vectors of $\hat{\mathbf{W}}$. Furthermore, since the set of the right singular vectors, i.e., $\{\mathbf{v}_j\}$, is an orthonormal basis set for $\Delta_{\mathbf{y}}$, we can write the gradient at the output as:

$$\Delta_{\mathbf{y}} = \sum_{j=1}^{n^2 d} \mathbf{v}_j \langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle.$$

Thus, we can compute the expected value of the norm of the gradients as:

$$\begin{aligned} \mathbb{E}[\|\Delta_{\mathbf{x}}\|_2^2] &= \sum_{j=1}^{n^2 c} \sigma_j^2 \mathbb{E}[\langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle^2], \\ \mathbb{E}[\|\Delta_{\mathbf{y}}\|_2^2] &= \sum_{j=1}^{n^2 d} \mathbb{E}[\langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle^2], \end{aligned}$$

where we use the fact that $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = 0$ for $i \neq j$ and $\mathbf{u}_j^T \mathbf{u}_j = \mathbf{v}_j^T \mathbf{v}_j = 1$ and the expectation is over the data population. We propose to preserve the norm of the gradient, i.e., $\mathbb{E}[\|\Delta_{\mathbf{x}}\|_2^2] = \mathbb{E}[\|\Delta_{\mathbf{y}}\|_2^2]$, by setting all the non-zero singular values to σ . It is easy to show that we can achieve this by setting

$$\sigma^2 = \frac{\sum_{j=1}^{n^2 d} \mathbb{E}[\langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle^2]}{\sum_{j, \sigma_j \neq 0} \mathbb{E}[\langle \Delta_{\mathbf{y}}, \mathbf{v}_j \rangle^2]}, \quad (3.7)$$

where the summation in the denominator is over the singular vectors \mathbf{v}_j corresponding to the nonzero singular values, i.e., $\sigma_j \neq 0$. The ratio in (3.7) is the ratio of expected energy of $\Delta_{\mathbf{y}}$, i.e. $\mathbb{E}[\|\Delta_{\mathbf{y}}\|_2^2]$, divided by the portion of energy that does not lie in the null space of $\hat{\mathbf{W}}$. We make the assumption that this ratio can be approximated by $\frac{n^2 d}{n^2 \min(d, c)}$. This assumption implies

that about $\frac{n^2 \min(d,c)}{n^2 d}$ of the total energy of $\Delta_{\mathbf{y}}$ will lie in the $n^2 \min(d, c)$ -dimensional subspace, corresponding to orthonormal basis set $\{\mathbf{v}_j | \sigma_j \neq 0\}$, of our $n^2 d$ -dimensional space. It is easy to notice that the assumption holds if the energy of $\Delta_{\mathbf{y}}$ is distributed uniformly among the directions in the basis set $\{\mathbf{v}_j\}$. But, since we are taking the sum over a large number of bases, it can also hold with high probability in cases where there is some variation in the distribution of energies along different directions. This is not a strict assumption in high dimensional spaces and we will investigate the practical relevance of this assumption in a real-world setting shortly. Thus, we can achieve norm preservation by setting all the nonzero singular values to $\sqrt{\frac{d}{\min(d,c)}}$. We can enforce this equality without using singular value decomposition. For that, we use the following theorem from [44]. This theorem states that the singular values of the convolution operator can be calculated by finding the singular values of the Fourier transform of the slices of the convolution kernels.

Theorem 3.3. (Theorem 6 from [44]) For any convolution kernel $\mathbf{K} \in \mathbb{R}^{k \times k \times d \times c}$ acting on an $n \times n \times d$ input, let $\hat{\mathbf{W}}$ be the matrix encoding the linear transformation computed by a convolutional layer parameterized by \mathbf{K} . Also, for each $u, v \in [n] \times [n]$, let $\mathbf{P}^{(u,v)} \in \mathbb{C}^{d \times c}$ be the matrix given by $\mathbf{P}_{i,j}^{(u,v)} = (\mathcal{F}_n(\mathbf{K}_{:, :, i, j}))_{u,v}$, where $\mathcal{F}_n(\cdot)$ is the operator describing an $n \times n$ 2D Fourier transform. Then, the set of singular values of $\hat{\mathbf{W}}$ is the union (allowing repetitions) of all the singular values of $\mathbf{P}^{(u,v)}$ matrices $\forall u, v$.

Proof. See [44]. □

Hence, to satisfy the condition (3.7), we can set all the nonzero singular values of $\mathbf{P}^{(u,v)}$ to $\sqrt{\frac{d}{\min(d,c)}}$ for all u and v . This can be done by finding the matrix $\hat{\mathbf{P}}^{(u,v)}$ that minimizes $\|\mathbf{P}^{(u,v)} - \hat{\mathbf{P}}^{(u,v)}\|_F^2$, such that $\hat{\mathbf{P}}^{(u,v)T} \hat{\mathbf{P}}^{(u,v)} = \frac{d}{\min(d,c)} \mathbf{I}$, where $\|\cdot\|_F$ denotes the Frobenius norm and \mathbf{I} is a $c \times c$ identity matrix. It can be shown that the solution to this problem is given by

$$\hat{\mathbf{P}}^{(u,v)} = \sqrt{\frac{d}{\min(d,c)}} \mathbf{P}^{(u,v)} (\mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)})^{-\frac{1}{2}}. \quad (3.8)$$

This is closely related to *Procrustes* problems, in which the goal is to find the closest orthogonal matrix to a given matrix [45]. Finding the inverse of the square root of product $\mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)}$ can be computationally expensive, specifically for large number of channels c . Thus, we exploit an iterative algorithm that computes the inverse of the square root using only matrix multiplications. Specifically, one can use the following iterations to compute $(\mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)})^{-\frac{1}{2}}$ [46]:

$$\begin{aligned} \mathbf{T}_k &= 3\mathbf{I} - \mathbf{Z}_k \mathbf{Y}_k, \\ \mathbf{Y}_{k+1} &= \frac{1}{2} \mathbf{Y}_k \mathbf{T}_k, \\ \mathbf{Z}_{k+1} &= \frac{1}{2} \mathbf{T}_k \mathbf{Z}_k, \end{aligned} \tag{3.9}$$

for $k = 0, 1, \dots$ and the iterators are initialized as:

$$\mathbf{Y}_0 = \mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)}, \mathbf{Z}_0 = \mathbf{I}.$$

It has been shown that \mathbf{Z}_k converges to $(\mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)})^{-\frac{1}{2}}$ quadratically [46]. Since the iterations only involve matrix multiplication, they can be implemented efficiently on GPUs.

Algorithm 1 Update rules for transition kernels at each iteration

Input: Convolution kernel \mathbf{K} at the current iteration

- 1: Perform the gradient descent step on the kernel \mathbf{K} .
 - 2: Calculate $\mathbf{P}^{(u,v)}$ for each $u, v \in [n] \times [n]$ as $\mathbf{P}_{i,j}^{(u,v)} = (\mathcal{F}_n(\mathbf{K}_{[:, :, i, j})))_{u,v}$.
 - 3: Compute $(\mathbf{P}^{(u,v)T} \mathbf{P}^{(u,v)})^{-\frac{1}{2}}$ using (3.9).
 - 4: Calculate $\hat{\mathbf{P}}^{(u,v)}$ using (3.8).
 - 5: Update \mathbf{K} using the inverse 2D Fourier transform of $\hat{\mathbf{P}}^{(u,v)}$.
-

Thus, to keep the convolution kernels norm preserving throughout the training, at each iteration, we compute the matrices $\mathbf{P}^{(u,v)}$ and set the nonzero singular values using (3.8). Algorithm 1 summarizes the operations performed at each iteration on the kernels of the regularized convolution layers. To keep the desired norm-preservation property after performing the gradient descent step, such as SGD, Adam, etc, the proposed scheme is used to re-enforce norm-preservation on the up-

dated kernel. In this manner, we can maintain norm-preservation, while updating the kernel during the training. Our experiments in Section 3.3 show that performing the proposed projection on the transition block of deep ResNets increases the training time by less than 8%. Also, since the number of transition blocks are independent of depth, the deeper the network gets, the computational overhead of the proposed modification becomes less significant. Figure 3.1(d) shows the diagram of the proposed transition block, where a regularized convolution layer, conv^* , is used to change the dimension. Hence, we are able to exploit a regular residual block with identity mapping, which is norm preserving.

Similar to [26], to take into the account the effect of a ReLU nonlinearity and to make a Conv-Relu layer norm-preserving, we just need to add a factor of $\sqrt{2}$ to the singular values and set them to $\sqrt{\frac{2d}{\min(d,c)}}$. Intuitively, the element-wise ReLU sets half of the units to zero on average, making the expected value of the energy of the gradient equal to $\mathbb{E}[\|\Delta_x\|_2^2] = \frac{1}{2} \sum_{j=1}^{n^2c} \sigma_j^2 \mathbb{E}[|\langle \Delta_y, v_j \rangle|^2]$. Therefore, to compensate this, we need to satisfy this condition:

$$\frac{1}{2} \sum_{j=1}^{n^2c} \sigma_j^2 \mathbb{E}[|\langle \Delta_y, v_j \rangle|^2] = \sum_{j=1}^{n^2d} \mathbb{E}[|\langle \Delta_y, v_j \rangle|^2]$$

It is also worthwhile to mention that since we are trying to preserve the norm of the backward signal, the variable n in Theorem 3.3 represents the size of feature map size at the output of the convolution.

To evaluate the effectiveness of the proposed projection, we design the following experiment. We perform the projection on the convolution layers of a small 3-layer network. The network consists of 3 convolutional layers, followed by ReLU non-linearity. To examine the gradient norm ratio for different number of input and output channels, the second layer is a 3×3 convolution with c input channels and d output channels. The first and third layers are 1×1 convolutions to change

the number of channels and to match the size of the input and output layers. Figure 3.2 shows the gradient norm ratio, i.e., $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}}\|_2$ to $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l}\|_2$, for different values of c and d at 10th training epoch on CIFAR-10, with and without the proposed projection. The values are averaged over 10 different runs.

It is evident that the proposed projection enhances the norm preservation of the Conv-ReLU layer, as it moves the gradient norm ratios toward 1. The only failure case is for networks with very small c and $c \ll d$. This is because, due to the smaller size of the space, our assumption that the energy of the signal in the n^2c dimensional subspace, corresponding to the n^2c non-zero singular values, is approximately $\frac{n^2c}{n^2d}$ of the total energy of the signal, is violated with higher probability. However, in more practical settings, where the number of channels is large and the assumption is held, the proposed projection performs as expected. This experiment illustrates the validity of our analysis as well as the effectiveness of the proposed projection for such practical scenarios. In the next section, we demonstrate the advantages of the proposed method for image classification task.

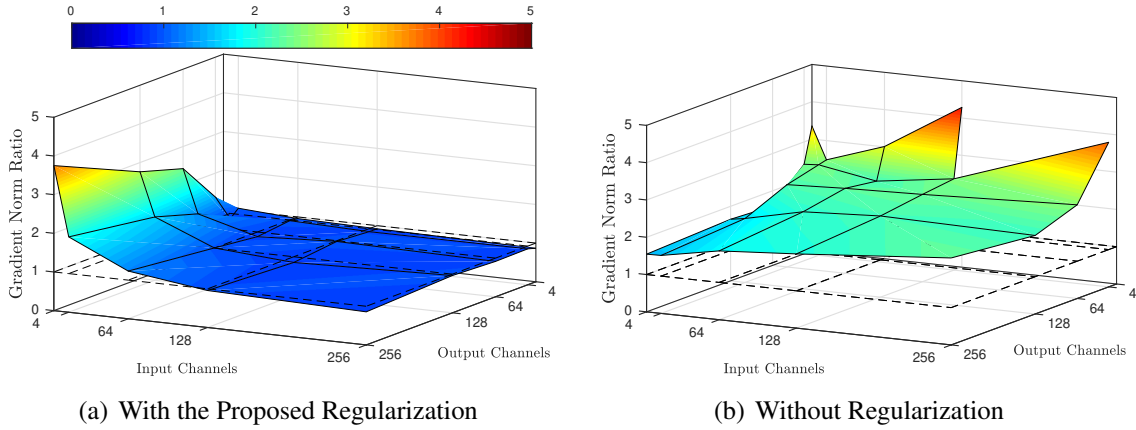


Figure 3.2: The ratio of gradient norm at output to gradient norm at input, i.e., $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}}\|_2$ to $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l}\|_2$, of a convolution layer for different number of input and output channels at 10th training epoch (a) with, and (b) without the proposed regularization on the singular values of the convolution.

3.3 Experiments

To validate our theoretical investigation, presented in Section 3.1, and to empirically demonstrate the behavior and effectiveness of the proposed modifications, we experimented with Residual Network (ResNet) and the proposed Procrustes Residual Network (ProcResNet) architectures on CIFAR10 and CIFAR100 datasets. Training and testing datasets contain 50,000 and 10,000 images of visual classes, respectively [47]. Standard data augmentation (flipping and shifting), same as [11, 12, 30], is adopted. Furthermore, channel means and standard deviations are used to normalize the images. The network is trained using stochastic gradient descent. The weights are initialized using the method proposed in [26] and the initial learning rate is 0.1. Batch size of 128 is used for all the networks. The weight decay is 10^{-4} and momentum is 0.9. The results are based on the top-1 classification accuracy.

Experiments are performed on three different network architectures: 1. **ResNet** contains one convolution layer, L residual blocks, three of which are transition blocks, and one fully connected layer. Each residual block consists of three convolution layers, as depicted in Figure 3.1(b) and Figure 3.1(c), resulting in a network of depth $3L + 2$. This is the same architecture as in [12]. 2. **ProcResNet** has the same architecture as ResNet, except the transition layers are modified, as explained in Section 3.2. In this design, 3 extra convolution layers are added to the network. However, we can use the first convolution layer of the original ResNet design to match the dimensions and only add two extra layers. This leads to a network of depth $3L + 4$. 3. **Plain** network is also same as ResNet without the skip connection in all the L residual blocks, as shown in Figure 3.1(e).

Furthermore, to decrease the computational burden of the proposed regularization, we perform the projection, as described in Section 3.2, every 2 iterations. This reduces the computation time significantly without hurting the performance much. In this setting, performing the proposed regularization increases the training time for ResNet164 about 7.6%. However, since we perform the

regularization only on three blocks, regardless of the depth, as the network becomes deeper the computational overhead becomes less significant. For example, implementing the same projections on ResNet1001 increases the training time by only 3.5%. This is significantly less computation compared to regularization using SVD, which leads to 53% and 23% training time overhead for ResNet164 and ResNet1001, respectively².

3.3.1 Norm-Preservation

In the first set of experiments, the behavior of different architectures is studied as the function of network depth. To this end, the ratio of gradient norm at output to gradient norm at input, i.e., $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}}\|_2$ to $\|\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l}\|_2$, is captured for all the residual blocks³, both transition and non-transition. Figure 3.3 shows the ratios for different blocks over training epochs. We ran the training for 100 epochs, without decaying the learning rate. Plain network (Figure 3.3.(g)) with 164 layers became numerically unstable and the training procedure stopped after 10 epochs.

Several interesting observations can be made from this experiment:

- This experiment emphasizes the fact that one needs more than careful initialization to make the network norm-preserving. Although the plain network is initially norm-preserving, the range of the gradient norm ratios becomes very large and diverges from 1, as the parameters are updated. However, ResNet and ProcResNet are able to enforce the norm-preservation during training procedure by using identity skip connection.
- As the networks become deeper, the plain network becomes less norm preserving, which leads to numerical instability, optimization difficulty, and performance degradation. On the contrary,

²An implementation of ProcResNet is provided here: <https://github.com/zaeemzadeh/ProcResNet>

³In Plain architecture, which does not have skip connections, the gradient norm ratio is obtained at the input and output of its building blocks as depicted in Figure 3.1(e).

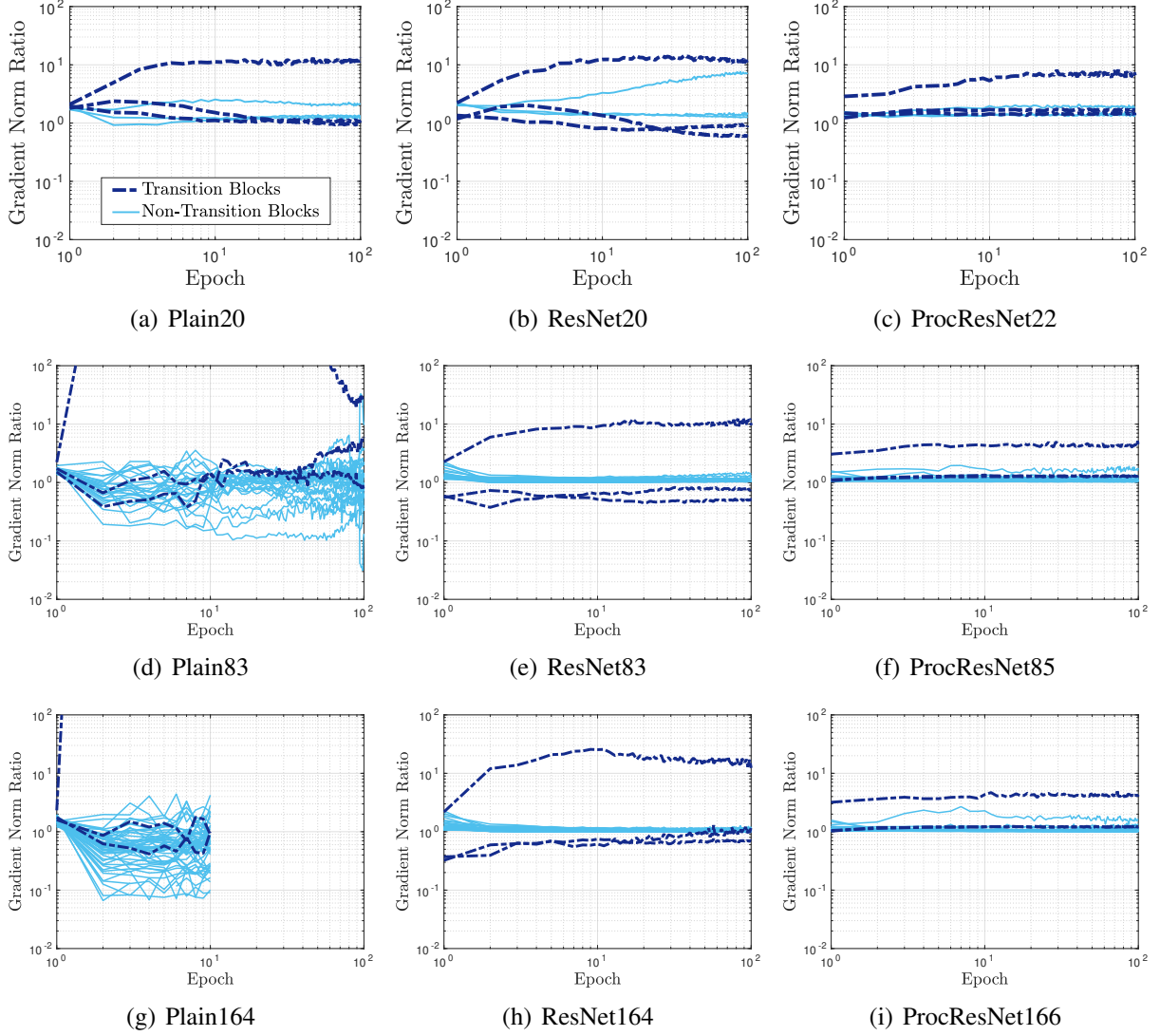


Figure 3.3: Training on CIFAR10. Gradient norm ratio over the first 100 epochs for transition blocks (blocks that change the dimension) and non-transition blocks (blocks that do not change the dimension). The darker color lines represent the transition blocks and the lighter color lines represent the non-transition blocks. The proposed regularization enhances the norm-preservation of the transition blocks effectively.

the non-transition blocks, the blocks with identity mapping as skip connection, of ResNet and ProcResNet become extra norm preserving. This is in line with our theoretical investigation for linear residual networks, which states that as we stack more residual blocks the network becomes extra norm-preserving.

- Comparing Plain83 (Figure 3.3(d)) and Plain164 (Figure 3.3(h)) networks, it can be observed that most of the blocks behave fairly similar, except one transition block. Specifically, in Plain83, the gradient norm ratio of the first transition block goes up to 100 in the first few epochs. But it eventually decreases and the network is able to converge. On the other hand, in Plain164, the gradient norm ratio of the same block becomes too large, which makes the network unable to converge. Hence, a single block is enough to make the optimization difficult and numerically unstable. This highlights the fact that it is necessary to enforce norm-preservation on all the blocks.
- In ResNet83 (Figure 3.3(e)) and ResNet164 (Figure 3.3(h)), it is easy to notice that only 3 transition blocks are not norm preserving. As mentioned earlier, due to multiplicative effect, the magnitude of the gradient will not be preserved because of these few blocks.
- The behaviors of ResNet and Plain architectures are fairly similar for depth of 20. This was somehow expected, since it is known that the performance gain achieved by ResNet is more significant in deeper architectures [11]. However, even for depth of 20, ProcResNet architecture is more norm preserving.
- In ProcResNet, the only block that is less norm preserving is the first transition block, where the 3 RGB channels are transformed into 64 channels. This is because, as we have shown in Figure 3.2, under such condition, where the number of input channels is very small, the assumption that energy of the gradient signal in the low-dimensional subspace, corresponding to the few non-zero singular values, is approximately proportional to the size of the subspace is violated

with higher probability.

- The ratios of the gradients for all networks, even the Plain network, are roughly concentrated around 1, while training is stable. This shows that some degree of norm preservation exists in any stable network. However, as clear in the Plain network, such biases of the optimizer is not enough and we need skip connections to enforce norm preservation throughout training and to enjoy its desirable properties. Furthermore, although the transition blocks of ResNet tend to converge to be more norm preserving, our proposed modification enforces this property for all the epochs, which leads to stability and performance gain, as will be discussed shortly.

This experiment both validates our theoretical arguments and clarifies some of the inner workings of ResNet architecture, and also shows the effectiveness of the proposed modifications in ProcResNet. It is evident that, as stated in Theorem 3.1, addition of identity skip connection makes the blocks increasingly extra norm-preserving, as the network becomes deeper. Furthermore, we have been able to enhance norm-preserving property by applying the changes proposed in Section 3.2.

3.3.2 *Optimization Stability and Learning Dynamics*

In the next set of experiments, numerical stability and learning dynamics of different architectures is examined. For that, loss and classification error, in both training and testing phases, are depicted in Figure 3.4. This experiment illustrates that how optimization stability of deep networks is improved significantly, and how it can be further improved by having norm preservation in mind during the design procedure.

As depicted in Figure 3.4, unlike the plain network, training error and loss curves corresponding to ResNet and ProcResNet architectures are consistently decreasing as the number of layers increases, which was the main motivation behind proposing residual blocks [11]. Moreover, Figure

3.4(a) and Figure 3.4(d) show that the plain networks have a poor generalization performance. The fluctuations in testing error shows that the points along the optimization path of the plain networks do not generalize well. This issue is also present, to a lesser extent, in ResNet architecture. Comparing Figure 3.4(h) and 3.4(b), we can see that the fluctuations are more apparent in deeper ResNet networks. However, in proposed ProcResNet architecture, the amplitude of the fluctuations is smaller and does not change as the depth of the network is increased. This indicates that ProcResNet architecture is taking a better path toward the optimum and has better generalization performance.

To quantify this, we repeated the training 10 times with different random seeds and measured the generalization gap, which is the difference between training and testing classification error, for the first 100 epochs. Table 3.1 shows the mean and max generalization gap, averaged over 10 different runs. This results indicate that generalization gap of ProcResNet is smaller. Furthermore, the generalization gap fluctuates far less significantly for ProcResNet, as quantified by the difference between mean generalization gap and maximum generalization gap.

Table 3.1: Mean and maximum generalization gap (%) during the first 100 epochs of training on CIFAR10 for different network architectures, averaged over 10 runs.

Depth	Plain		ResNet		ProcResNet	
	mean	max	mean	max	mean	max
20	6.7	20.0	5.5	23.1	2.3	8.3
83	7.5	30.1	5.1	12.5	2.0	7.7
164	-	-	5.2	18.7	3.3	8.7

The implication of this is that by modifying only a few blocks in an extremely deep network, it is possible to make the network more stable and improve the learning dynamics. This emphasizes the utmost importance of norm-preservation of all blocks in avoiding optimization difficulties of very deep networks. Moreover, this sheds light on the reasons why architectures using residual blocks,

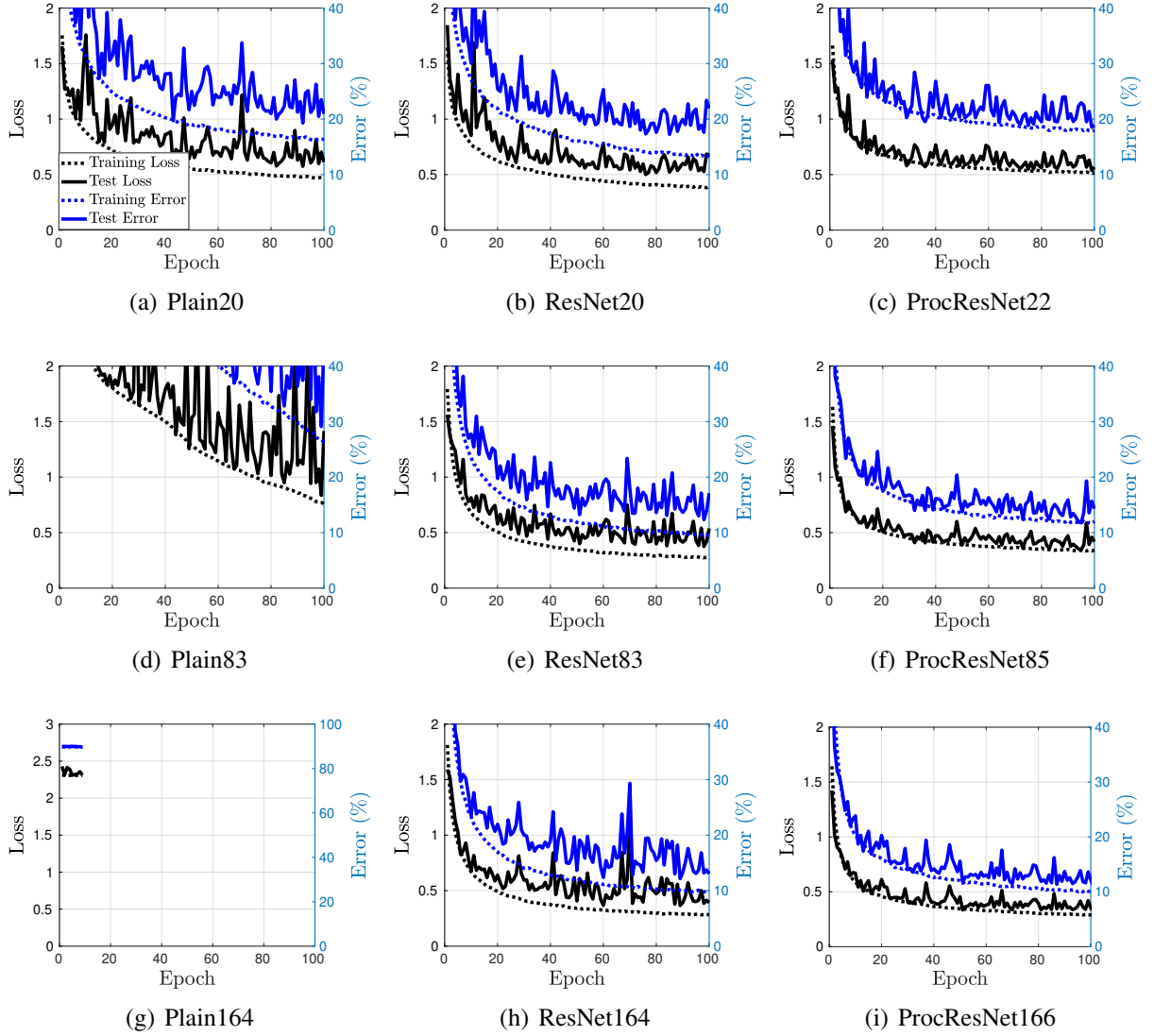


Figure 3.4: Loss (black lines) and error (blue lines) during training procedure on CIFAR10. Solid lines represent the test values and dotted lines represent the training values. This experiments shows how the residual connections enhance the stability of the optimization and how the proposed regularization enhances the stability even further.

or identity skip connection in general, perform so well and are easier to optimize.

3.3.3 Classification Performance

In this section, we show the impact of the proposed norm-preserving transition blocks on the classification performance of ResNet. Table 3.2 compares the performance of ResNet and its EraseReLU version, as proposed in [48], with and without the proposed transition blocks. The results for standard ResNet are the best results reported by [12] and [48] and the results of ProcResNet are obtained by making the proposed changes to standard ResNet implementation.

Table 3.2 shows that the proposed network performs better than the standard ResNet. This performance gain comes with a slight increase the number of parameters (under 1%) and by changing only 3 blocks. The total number of residual blocks for ResNet164 and ResNet1001 are 54 and 333, respectively. Furthermore, Figure 3.5 compares the parameter efficiency of ResNet and ProcResNet architectures. The results indicate that the proposed modification can improve the parameter efficiency significantly. For example, ProcResNet274 (with 2.82M parameter) slightly outperforms ResNet1001 (with 10.32M parameters). This translates into about 4x reduction in the number of parameters to achieve the same classification accuracy. This illustrates that we are able to improve the performance by changing a tiny portion of the network and emphasizes the importance of norm-preservation in the performance of neural networks.

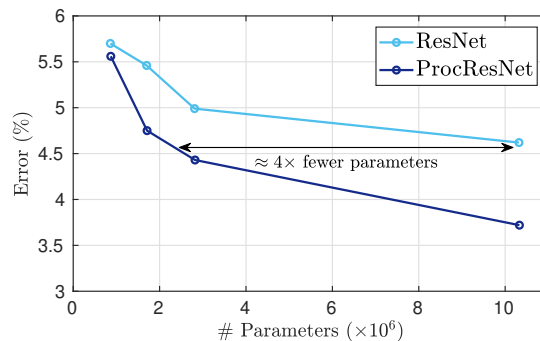


Figure 3.5: Comparison of the parameter efficiency on CIFAR10 between ResNet and ProcResNet.

Finally, Table 3.3 investigates the impact of changing the architecture, i.e., moving the convolution layer from the skip connection to before the skip connection, and performing the proposed regularization, separately. Each of these design components have positive impact on the performance of the network, as both of them enhance the norm preservation of the transition block, which further highlights the impact of norm preservation on the performance of the network.

Table 3.2: Performance of different methods on CIFAR-10 and CIFAR-100 using moderate data augmentation (flip/translation). The modified transition blocks in ProcResNet can improve the accuracy of ResNet significantly.

Architecture	Setting	# Params	Depth	Error (%)	
				CIFAR10	CIFAR100
ResNet [12]	pre-activation	1.71M	164	5.46	24.33
		10.32M	1001	4.62	22.71
	ErasedReLU[48]	1.70M	164	4.65	22.41
		10.32M	1001	4.10	20.63
ProcResNet	pre-activation	1.72M	166	4.75	22.61
		10.33M	1003	3.72	19.99
	ErasedReLU[48]	1.72M	166	4.53	21.91
		10.33M	1003	3.42	18.12

Table 3.3: Ablation study on ResNet with 164 layers on CIFAR100.

Transition Block	Projection	Error (%)
Original	No	24.33
Modified	No	23.06
Modified	Yes	22.61

3.4 Conclusions

This chapter theoretically analyzed building blocks of residual networks and demonstrated that adding identity skip connection makes the residual blocks norm-preserving. This means that the

norm-preservation is enforced during the training procedure, which makes the optimization stable and improves the performance. This is in contrast to initialization techniques, such as [36], which ensure norm-preservation only at the beginning of the training. Our experiments validated our theoretical investigation by showing that (i) identity skip connection results in norm preservation, (ii) residual blocks become extra norm-preserving as the network becomes deeper, and (iii) the training can become more stable through enhancing the norm preservation of the network. Our proposed modification of ResNet, Procrustes ResNet, enforces norm-preservation on the transition blocks of the network and is able to achieve better optimization stability and performance. For that we proposed an efficient regularization technique to set the nonzero singular values of the convolution operator, without performing singular value decomposition. Our findings can be seen as design guidelines for very deep architectures. By having norm-preservation in mind, we will be able to train extremely deep networks and alleviate the optimization difficulties of such networks.

CHAPTER 4: OUT-OF-DISTRIBUTION DETECTION USING UNION OF 1-DIMENSIONAL SUBSPACES

The goal of out-of-distribution (OOD) detection is to handle the situations where the test samples are drawn from a different distribution than the training data. In this work, we claim that we can improve the OOD detection performance by *constraining* the representation of in-distribution (ID) samples in the feature space. Particularly, if we embed the training samples such that the feature vectors belonging to each known class lie on a 1-dimensional subspace, OOD samples can be detected more robustly with higher probability, compared to a class-conditional non-degenerate Gaussian embeddings. Such a *union of 1-dimensional subspaces* representation provides us with two main advantages. First, due to compact representation in the feature space, OOD samples are less likely to occupy the same region as the known classes. In other words, a random vector in a high-dimensional space lies on a specific 1-dimensional line with probability 0. Second, we show that the first singular vector of a 1-dimensional subspace is a robust representative of its samples. We exploit these two desirable features and reject samples as OOD, if they occupy the region corresponding to the training samples with probability 0. This region is identified by the set of the first singular vectors of the training classes. To estimate the probability, we use Monte Carlo sampling techniques used in Bayesian deep learning such as [49, 50]¹.

Our work is primarily motivated by the rich literature of spectral methods in signal processing and machine learning. Spectral techniques have been proven to be very effective for different tasks such as robust estimation and detection [52, 53], learning mixture models [54], representative selection [55], and defense against backdoor attacks [56]. We are also inspired by the OOD detection method

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-Distribution Detection Using Union of 1-Dimensional Subspaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, © 2021 IEEE [51]

proposed in [19], in which authors use the ID feature vectors to estimate their distribution and to detect OOD samples. In contrast, we engineer the distribution of ID feature vectors to minimize the error probability, without knowing the distributions of OOD samples, and enforce our desired distribution on the feature vectors. Our proposed method does not need extra information or a subset of OOD examples for hyperparameter tuning or validation. This is in contrast to many existing methods that use some subset of the OOD samples, either during validation [14, 15, 19, 57], or even during training [58, 59]. Despite improving the results, the availability of such extra information is questionable in many real-world applications. Furthermore, our technique can be easily deployed on many existing frameworks and different modalities, e.g. images, videos, etc. In summary, this chapter makes the following contributions:

- We demonstrate that if feature vectors lie on a union of 1-dimensional subspaces, the OOD samples can be robustly detected with high probability and we show how we can impose such constraint on the ID feature vectors (Section 4.2);
- We propose a new OOD detection test, which exploits the first singular vector of the feature vectors extracted from the training set, in conjunction with MC sampling (Section 4.3);
- Our framework does not have hyperparameters, does not need extra information, and can be easily applied to existing methods with minimal change. Furthermore, the proposed method can be applied to different domains such as images and videos.

4.1 Related Work

The problem of detecting outliers and anomalies in the data has been extensively studied in machine learning and signal processing communities and is closely related to outlier detection, a topic that has been greatly studied both in the supervised [60] and unsupervised [61] settings. The liter-

ature in this area is sizable. Thus, we mainly focus on the recent deep learning approaches. These methods either estimate the distribution of ID samples [19, 57] or use a distance metric between the test samples and ID samples to detect OOD samples [13, 14].

Many of the existing approaches employ the OOD datasets during training [58, 59] or validation steps [14, 15, 19, 57, 62, 63]. For instance, in [59], the network is fine-tuned during the training to increase the distance between ID and OOD distributions. Other interesting methods, such as [14, 15, 19], apply a perturbation on each sample at test time to exploit the robustness of their network in detecting ID samples. However, they use part of the OOD samples to fine-tune the perturbation parameters. On the other hand, methods that rely on generative models or autoencoders, such as [57], also require hyperparameter tuning for loss terms, regularization terms, and/or latent space size. Authors in [64] propose to use extra supervision, in particular several word embeddings, to construct a better latent space and to detect OOD samples more accurately. A table summarizing the prior work and how they leverage extra information is provided in Appendix B. Having access to extra information certainly helps with the performance. However, it can be argued that OOD detectors should be completely agnostic of unknown distributions, which is a more realistic scenario in the wild. On the other hand, only a few approaches, such as [13, 18, 65–67], do not require the OOD samples neither during training nor validation. For instance, Hendricks and Gimpel [13] show how the softmax layer can be used to detect OOD samples, when its prediction score is below a threshold. In [18], the authors rely on reconstructing the samples to produce a discriminative feature space. However, methods that rely on either reconstruction or generation [18, 57, 65] do not perform well in scenarios where sample generation or reconstruction is more difficult, such as large-scale datasets or video classification.

4.2 Union of 1-dimensional Subspaces for Out-of-Distribution Detection

Given a training dataset consisting of N sample-label pairs belonging to L known classes, our goal is to train a neural network such that at the test time it can be determined if an unlabeled sample is an out-of-distribution sample (not belonging to any of the L known classes) or not. We are particularly interested in the scenarios where OOD samples are not available. Thus, we do not use OOD samples during training or validation. We argue that OOD detection performance can be improved if the feature vectors from the known classes lie on a *union of 1-dimensional subspaces*. In short, such embedding has two main properties that we can take advantage for OOD detection: (i) Due to the compactness of ID samples in the feature space, OOD samples can be detected with higher probability, compared to conventional class-conditional non-degenerate Gaussian embeddings, and (ii) First singular vector of the samples in each class can be used as a robust representative of that class and can be effectively employed to distinguish between the ID and OOD samples. Below, we discuss each of these advantages in more details.

Distribution-agnostic minimization of error probability: Computing the error probability for OOD detection is a difficult task to carry out. This is due to the fact that, by definition, we do not have much information about the probability distribution of the OOD samples. However, it can be shown that the probability of error can be minimized by making the distribution of the known classes as compact as possible. Specifically, consider the binary classification problem of distinguishing between the OOD samples and samples from one of the known classes, following multivariate Gaussian distributions with different means and covariance matrices $\mathcal{N}(\boldsymbol{\mu}_o, \boldsymbol{\Sigma}_o)$ and $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, respectively. It has been shown [68] that the classification error probability p_e can be upper bounded by: $p_e \leq \sqrt{p_i p_o} e^{-B}$, where p_i and p_o are the probability of samples belonging to

the known class and OOD samples, respectively. B is the Bhattacharyya distance defined as:

$$B = \frac{1}{8} \Delta^T \left(\frac{\Sigma_i + \Sigma_o}{2} \right)^{-1} \Delta + \frac{1}{2} \ln \left(\frac{\det(\frac{\Sigma_i + \Sigma_o}{2})}{\sqrt{\det(\Sigma_i) \det(\Sigma_o)}} \right),$$

where $\Delta = \mu_i - \mu_o$ is the distance between the means of the two distributions. The first term in B represents the Mahalanobis distance between μ_i and μ_o , using $\frac{\Sigma_i + \Sigma_o}{2}$ as the covariance matrix. The second term is a measure of compactness of the distributions. The larger the $\det(\Sigma_i)$ is, the more its corresponding samples are spread out. Thus, even without any knowledge about μ_o , Σ_o , p_i , and p_o , one can increase B by making $\mathcal{N}(\mu_i, \Sigma_i)$ as compact as possible. In the extreme case, where the samples lie on a perfect 1-dimensional subspace, error probability will be 0, unless the OOD feature vectors have the exact same distribution as the known class. To demonstrate this in further details, consider the following toy examples:

Example 1: Let $\Sigma_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\Sigma_i = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}$, $\epsilon \ll 1$, meaning that the ID samples occupy an almost 1-dimensional subspace of the 2-dimensional space. In this example, the second term in above equation becomes $\ln(\frac{1+\epsilon^2}{2\epsilon})$, which approaches infinity as $\epsilon \rightarrow 0$, making p_e very small. This is true even if $\mu_i = \mu_o$.

Example 2: Let $\Sigma_o = \Sigma_i = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2 \end{bmatrix}$, $\epsilon \ll 1$, $\mu_i = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \end{bmatrix}$, $\mu_o = \begin{bmatrix} \mu_{o1} \\ \mu_{o2} \end{bmatrix}$, i.e., ID and OOD samples have the same degenerate covariance matrix. In this case, the second term becomes 0, but the first term, which is the Mahalanobis distance between the mean vectors, is $\frac{1}{8}[(\mu_{i1} - \mu_{o1})^2 + \frac{1}{\epsilon^2}(\mu_{i2} - \mu_{o2})^2]$. If $\epsilon \rightarrow 0$, p_e approaches 0, unless $(\mu_{i2} - \mu_{o2})^2 \rightarrow 0$ as well. This means that if the means of the distribution have some mismatch along the degenerate direction, even though very small, OOD samples can be detected with very small p_e .

Thus, by enforcing the ID feature vectors to lie on 1-dimensional subspaces, we can detect slight mismatches between the distribution of the OOD samples in feature space and the distribution of

ID samples, which leads to better OOD detection.

First singular vector as a robust representative: In the context of robust statistics, the first singular vector has been shown to be a great tool to define robust mean and covariance estimators [52]. In addition, the first singular vector has been used to select the representatives of the class[55]. It can be shown that the first singular vector is robust to perturbations and noise. Let \mathbf{X}_l denote an $M \times N$ matrix containing N M -dimensional *feature vectors* belonging to class l . Furthermore, consider the autocorrelation matrix of the class l defined as $\mathbf{C}_l = \mathbf{X}_l \mathbf{X}_l^T$. Eigenvectors and eigenvalues of \mathbf{C}_l are the left singular vectors and the square of singular values of \mathbf{X}_l , respectively. Adding noise or adding a new noisy column in \mathbf{X}_l perturbs \mathbf{C}_l , without changing its dimensions. To quantify the sensitivity of eigenvectors of \mathbf{C}_l against perturbations, we use the following Lemma.

Lemma 4.1. (from [55]) Assume square matrix \mathbf{C} and its spectrum $[\lambda_i, \mathbf{v}_i]$. Then, $\|\partial \mathbf{v}_i\|_2 \leq \sqrt{\sum_{j \neq i} \frac{1}{(\lambda_i - \lambda_j)^2}} \|\partial \mathbf{C}\|_F$, where $\|\cdot\|_F$ denotes Frobenius norm and the partial derivative is taken with respect to any scalar variable.

If we take the partial derivative with respect to an entry in \mathbf{C} , we can see that the sensitivity of the i^{th} spectral component, \mathbf{v}_i , to perturbations in \mathbf{C} , is inversely related to the gap between its corresponding eigenvalue λ_i and other eigenvalues $\lambda_j, j \neq i$. Therefore, we can define the sensitivity coefficient of the i^{th} eigenvector of a square matrix as $s_i \triangleq \sqrt{\sum_{j \neq i} \frac{1}{(\lambda_i - \lambda_j)^2}}$. In general, the first singular component \mathbf{v}_1 is the least sensitive direction to the perturbations. This is because, in many scenarios, the gap between consecutive eigenvalues is decreasing (see [69] and references therein), which leads to $s_1 < s_i, \forall i \geq 2$. However, we can further increase the robustness, by embedding the ID feature vectors onto a union of 1-dimensional subspaces. Since the singular values represent the amount of energy concentrated along their corresponding singular vector, if almost all of the energy of the data points in each class is concentrated along its corresponding

first singular vector, we will have large λ_1 and small $\lambda_i, i \geq 2$ for all the classes. Therefore, if the feature vectors belonging to the same class lie on a 1-dimensional subspace, we can use the first singular vector of X_l as a robust representative of the class subspace in the feature space and to reject outliers, as shown in Section 4.3.

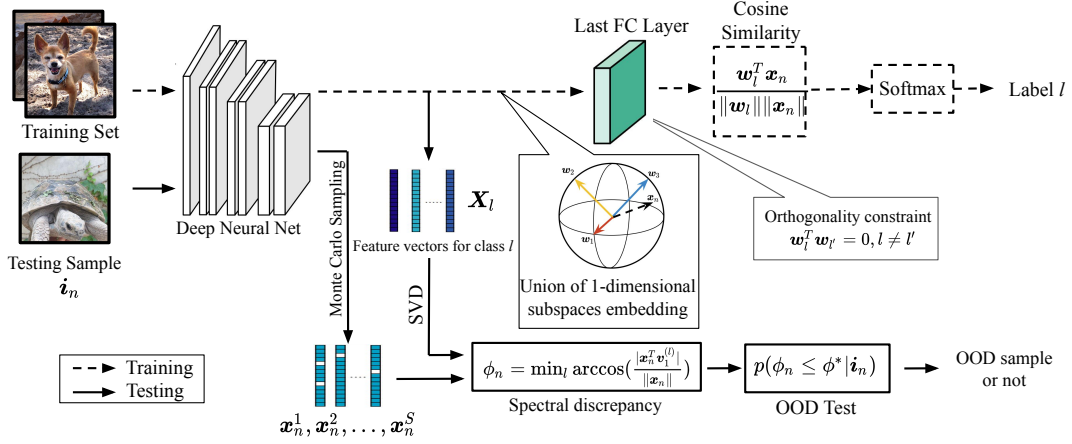


Figure 4.1: Overall architecture of the proposed framework. A neural network (e.g., WideResnet28) maps the input onto a feature space. Then, the cosine similarities between the extracted feature x_n and the class vectors w_l are used to compute the class membership probabilities. w_l s are set to predefined orthonormal vectors and are not updated during training. This leads to the desired embedding, union of uncorrelated 1-dimensional subspaces. At test time, the cosine similarity between the test samples and the first singular vector corresponding to each class is used to distinguish between the ID and OOD samples.

4.2.1 Enforcing the Structural Constraints

Intraclass Constraint: We can make the feature vectors for each known class to lie on a 1-dimensional subspace by employing cosine similarity. This can be achieved by modifying the softmax function to predict the membership probability using $p_{ln} = \frac{e^{|\cos(\theta_{ln})|}}{\sum_l e^{|\cos(\theta_{ln})|}}$, where p_{ln} is the probability of membership of feature vector n in class l and $\cos(\theta_{ln}) = \frac{w_l^T x_n}{\|w_l\| \|x_n\|}$ is the cosine similarity between the learned feature vector x_n and the weights of the last fully connected layer corresponding to class l , i.e., w_l . Note that, unlike other methods which employ angular margin

[70, 71], we use the absolute value of the cosine similarity to compute the class memberships. This is due to the fact that the subspace membership, and therefore the class membership, does not change if a vector is multiplied by -1 . By employing such activation function, the feature vectors of each class are aligned to its corresponding weight vector \mathbf{w}_l . In other words, class l forms a 1-dimensional subspace along the direction of \mathbf{w}_l in the feature space. Therefore the final loss function to be minimized is defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N -\log\left(\frac{e^{|\cos(\theta_n^*)|}}{\sum_l e^{|\cos(\theta_{ln})|}}\right), \quad (4.1)$$

where θ_n^* is angle between the n^{th} feature vector and the weight vector corresponding to its true label.

Interclass Constraint: By using the absolute cosine similarity as the classification criteria, we can ensure the feature vectors are angularly distributed in the space and form a union of 1-dimensional subspaces. To boost the interclass separation of the known classes, we need to decrease the interclass similarity, in terms of cosine similarity. Minimum interclass cosine similarity can be enforced by ensuring that \mathbf{w}_l are orthogonal to each other. We achieve this by simply initializing the weight matrix with orthonormal vectors, as described in [72], and freezing them during the training. Orthogonal initialization requires that $M > L$, which is often the case in practice (feature space dimension is larger than number of classes). In other words, the feature extractor, i.e., the deep neural network, is trained such that it can map each input sample in class l onto a predefined 1-dimensional subspace represented by the direction of \mathbf{w}_l .

Figure 4.1 shows the overall architecture of the proposed framework. The neural network maps the input sample onto a low-dimensional space, where the known classes are represented by a set of orthonormal vectors. The cosine similarity between the extracted feature from the n^{th} input sample, \mathbf{x}_n , and the vector corresponding to the class subspace, \mathbf{w}_l , is used to determine the class

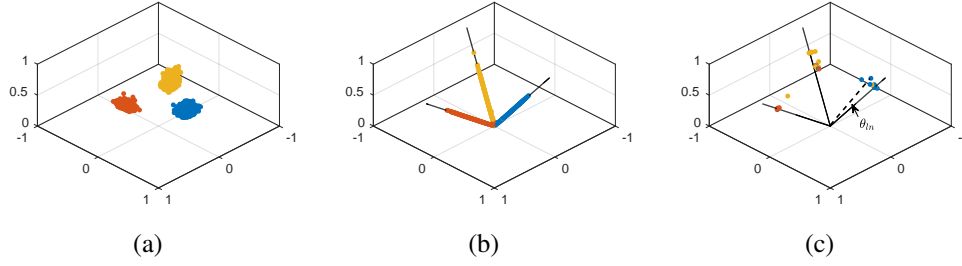


Figure 4.2: 3-dimensional representation of the features belonging to the first 3 classes of CIFAR10 training set, extracted from WideResNet with and without the proposed embedding: (a) features extracted from a plain WideResnet, (b) features extracted after enforcing the proposed embedding, and (c) same as (b) after ℓ_2 -normalizing the feature vectors. The solid lines represent the direction of the first singular vector corresponding to each class. All the figures contain 3,000 feature vectors.

membership probability and therefore the label. Figure 4.2 demonstrates the effectiveness of the proposed framework in enforcing the desired embedding. It shows a 3-dimensional embedding, obtained by PCA, of the feature vectors belonging to the first 3 classes of CIFAR10. The neural network, WideResnet28, is trained on all the classes of CIFAR10 with and without enforcing the proposed structural constraints. Figure 4.2(a) shows that the feature vectors belonging to each class extracted from a plain WideResnet have a fairly isometric Gaussian structure, meaning that they are spread out in different direction uniformly. On the other hand, as shown in Figure 4.2(b), the feature vectors extracted from the same network trained using our proposed technique lie on a union of 1-dimensional subspaces. We also show the ℓ_2 -normalized feature vectors in Figure 4.2(c) to remove the scale of the feature vectors and emphasize the angle between each vector and the singular vector corresponding to its class.

4.3 Out-of-distribution Detection Test

If the feature vectors belonging to the known classes lie on a union of 1-dimensional subspaces, their corresponding region in the feature space has no volume. Thus, the probability of OOD

samples being in the region corresponding to any of the known classes, which is the probability of false negative p_{fn} , is zero. This can be seen using the Bhattacharyya bound, discussed in Section 4.2, $p_e = p_o p_{fn} + p_i p_{fp} \leq \sqrt{p_i p_o} e^{-B}$. Therefore, if we make the known classes occupy a tiny region with no volume in the space, we will have $B \rightarrow \infty$ and $p_{fn} \rightarrow 0$. We use this property to classify samples as OOD if they lie inside the region corresponding to any of the known classes with probability 0. More specifically, given an input instance \mathbf{i}_n and corresponding feature vector \mathbf{x}_n , this probability can be estimated using the singular vectors of each class as $p(\phi_n \leq \phi^* | \mathbf{i}_n)$, where ϕ_n is defined as:

$$\phi_n = \min_l \arccos\left(\frac{|\mathbf{x}_n^T \mathbf{v}_1^{(l)}|}{\|\mathbf{x}_n\|}\right), \quad (4.2)$$

which is the minimum angular distance of the test feature vector \mathbf{x}_n , from the first singular vector of any of the classes. We name this measure as *spectral discrepancy*. ϕ^* is a critical spectral discrepancy and defines the region belonging to the known classes. Smaller values of ϕ^* corresponds to more compact regions. In the extreme case of $\phi^* = 0$, the input instance \mathbf{i}_n is detected as OOD, if it does not have the exact same direction as one of the singular vectors. It is worthwhile to mention that in the ideal case, the first singular vector of class l , $\mathbf{v}_1^{(l)}$, would be the same as \mathbf{w}_l . However, in practice, the first singular vector is a better representative of the subspace after training, as training feature vectors may not perfectly align with \mathbf{w}_l . $\mathbf{v}_1^{(l)}$ can be computed using the extracted features from training ID samples of class l . Time complexity order of computing the first singular vector is linear w.r.t both the number and the dimensions of the feature vectors [73, 74]. To estimate $p(\phi_n \leq \phi^* | \mathbf{i}_n)$, we employ Monte Carlo sampling. Specifically:

$$p(\phi_n \leq \phi^* | \mathbf{i}_n) = \int_0^{\phi^*} p(\phi_n | \mathbf{i}_n) d\phi_n \approx \frac{1}{S} \sum_{s=1}^S \mathbb{I}(\phi_n^s < \phi^*), \quad (4.3)$$

where S is the number of the Monte Carlo samples and ϕ_n^s is the spectral discrepancy of the s^{th} Monte Carlo sample, given input instance \mathbf{i}_n . Furthermore, $\mathbb{I}(\cdot)$ is the indicator function that takes

value 1 if $\phi_n^s < \phi^*$ and 0 otherwise. To obtain the samples, we can use the methods proposed for approximate Bayesian inference in [49, 50]. ϕ^* is the decision parameter, which can be set to achieve a problem-specific precision and/or recall requirements using different methods such as [75] or by using the training set (as will be discussed in Section 4.4).

Figure 4.3 demonstrates the effectiveness of employing spectral discrepancy in distinguishing between ID and OOD samples. Similar to Figure 4.2, this figure shows a 3-dimensional representation of the features that are close to the first 3 classes of the CIFAR10, meaning that the classifier would classify them as one of these classes. The first two subfigures show the features extracted from a plain WideResNet. Comparing ID samples (Figure 4.3(a)) with OOD samples (Figure 4.3(b)), it is clear that both ID and OOD samples follow a very similar structure, which makes OOD detection more difficult. On the other hand, the last two subfigures illustrate the ℓ_2 -normalized features extracted from the WideResNet trained using our proposed embedding. Comparing the ID (Figure 4.3(c)) and OOD (Figure 4.3(d)) samples, most of the OOD samples have larger angular distance to their closest singular vector, compared to the ID samples, which can be exploited to detect them more accurately. A quantitative evaluation of this example, including the histogram of spectral discrepancies for ID and OOD samples, is provided in Section 4.4 (e.g., Figure 4.4). Furthermore, an algorithmic description of the training and testing phases of our proposed method is provided in Appendix B.

4.4 Experiments

Datasets: We train the WideResNet model on CIFAR-10 and CIFAR-100 [47] datasets, which consist of 50,000 images for training and 10,000 images for testing, with an image size of 32×32 . The testing set is used as the ID testing samples. Similarly to prior work [14, 19, 66], for the OOD testing samples, we use the following datasets: (i) **TinyImagenet:** The Tiny ImageNet dataset

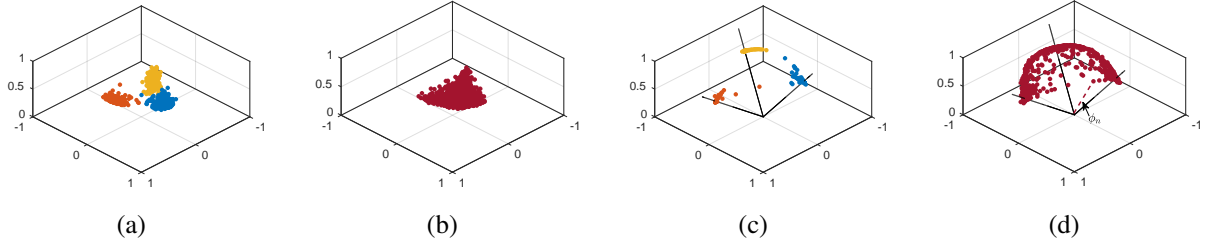


Figure 4.3: 3-dimensional representation of the features extracted from a plain WideResNet and the same network with our proposed embedding. (a) ID features extracted from plain network, (b) OOD features extracted from plain network, (c) ID features extracted using our embedding, and (d) OOD features extracted using our embedding. The solid lines represent the direction of the first singular vector corresponding to each class. OOD samples, extracted using our embedding, have larger angular distance to their closest singular vector. All the figures contain 3000 samples.

Table 4.1: A comparison of OOD detection results, in terms of F1-score, for different ID and OOD datasets. \dagger represents the results achieved by our re-run of the publicly available codes. The bottom section summarizes the performance of the methods that use a subset of OOD samples for hyperparameter tuning, such as finding the best perturbation magnitude. Our method does not have any parameters to be tuned.

ID dataset	CIFAR10				CIFAR100			
OOD dataset	TINc	TINr	LSUNc	LSUNr	TINc	TINr	LSUNc	LSUNr
SoftMax Pred. [13] \dagger	0.803	0.807	0.794	0.815	0.683	0.683	0.664	0.693
Counterfactual [65]	0.636	0.635	0.650	0.648	-	-	-	-
CROSR [18]	0.733	0.763	0.714	0.731	-	-	-	-
OLTR [66] \dagger	0.860	0.852	0.877	0.877	0.746	0.721	0.753	0.747
Ours	0.930	0.936	0.962	0.961	0.810	0.860	0.769	0.886
Methods that use OOD samples for validation and hyperparameter tuning.								
ODIN [14] \dagger	0.902	0.926	0.894	0.937	0.834	0.863	0.828	0.875
Mahalanobis [19] \dagger	0.985	0.969	0.985	0.975	0.974	0.944	0.963	0.952

consists of 10,000 test images of size 36×36 belonging to 200 different classes, which are sampled from the original 1,000 classes of ImageNet [5]. As in [14, 15] we construct two datasets from TinyImagenet: TinyImagenet-crop (TINc) and TinyImagenet-resize (TINr), by either randomly cropping or downsampling each image to a size of 32×32 . (ii) **LSUN**: LSUN [76] consists

of 10,000 test images from 10 different scene categories. Like before, we randomly crop and downsample the LSUN test set to construct two datasets LSUN-crop (LSUNc) and LSUN-resize (LSUNr).

Evaluation Metrics: We evaluate the OOD detection performance using the following metrics: **FPR at 95% TPR** indicates the false positive rate (FPR) at 95% true positive rate (TPR). **Detection Error** indicates the minimum misclassification probability. It is computed by the minimum misclassification rate over all possible values of ϕ^* . **AUROC**, defined as the Area Under the Receiver Operating Characteristic curve, is computed as the area under the FPR against TPR curve. **AUPR In** is computed as the area under the precision-recall curve. For AUPR In, ID images are treated as positive. **AUPR Out** is similar to the metric AUPR-In. Opposite to AUPR In, OOD images are treated as positive. **F1 Score** is the maximum average F1 score over all possible critical spectral discrepancy values ϕ^* .

Table 4.2: Performance of the proposed framework for distinguishing ID and OOD test set data for the image classification task, using a WideResnet with depth 28 and width 10. \uparrow indicates larger value is better and \downarrow indicates lower value is better. All the methods use the same network architecture.

Training dataset	OOD dataset	FPR at 95% TPR \downarrow	Detection Error \downarrow	AUROC \uparrow	AUPR In \uparrow	AUPR Out \uparrow
		Softmax. Pred. [13]/OLTR [66]/ Ours				
CIFAR10	TINc	38.9/25.6/ 9.0	21.9/14.8/ 6.8	92.9/91.3/ 98.1	92.5/93.2/ 98.2	91.9/88.3/ 98.1
	TINr	45.6/28.8/ 7.6	25.3/15.8/ 6.2	91.0/90.3/ 98.5	89.7/92.3/ 98.6	89.9/87.1/ 98.4
	LSUNc	35.0/21.3/ 2.8	20.0/13.0/ 3.7	94.5/92.9/ 99.4	95.1/94.4/ 99.4	93.1/90.8/ 99.4
	LSUNr	35.0/21.7/ 3.4	20.0/13.2/ 3.8	93.9/92.6/ 99.3	93.8/94.4/ 99.4	92.8/90.0/ 99.3
CIFAR100	TINc	66.6/63.8/ 41.7	35.8/29.0/ 18.9	82.0/77.4/ 88.6	83.3/78.7/ 89.1	80.2/74.4/ 87.0
	TINr	79.2/72.9/ 29.42	42.1/32.1/ 14.2	72.2/73.1/ 93.7	70.4/73.8/ 94.0	70.8/69.8/ 93.8
	LSUNc	74.0/59.2/ 38.8	39.5/29.1/ 13.9	80.3/76.9/ 93.8	83.4/80.0/ 93.6	77.0/72.9/ 93.1
	LSUNr	82.2/61.9/ 20.3	43.6/29.2/ 11.3	73.9/77.0/ 95.7	75.7/79.2/ 96.0	70.1/73.3/ 95.7

We deploy WideResNet with depth 28 and width 10 as the neural network architecture. The network parameters are set as the original implementations in [77, 78], except the last layer, which is modified as discussed in Section 4.2. At the test time, unless otherwise stated, we draw 50 Monte

Carlo samples to estimate $p(\phi_n \leq \phi^*)$ and to detect the OOD samples. To draw MC samples for the image classification task, we employ the SWAG-Diag method proposed in [49]. Other uncertainty estimation methods such as [50, 79–81] can also be used to estimate the uncertainty in conjunction with our proposed method. Additional training details are provided in Appendix B.

Table 4.1 compares our results with recent OOD detection techniques in terms of F1-score. As denoted in the table, we use the code provided by the authors from most of the baselines to generate the results under a fair setting, i.e., same architecture, same datasets, and same metrics. For [18, 65], we provide the results reported by the authors, as these methods rely on reconstruction and/or generation of samples and the same architecture cannot be used. In addition, since these methods only report their performance using F1-score, we also use this metric for all the methods. Our proposed method is able to consistently outperform the competing methods over different datasets, and is the closest competitor to the techniques that use OOD sample for validation. Table 4.2 compares the performance of our proposed solution with two of the more competitive baselines over different metrics, using the same network architecture for all the methods. Our results are consistent over different OOD datasets and different metrics, meaning that our method can perform well for different types of OOD samples, without any hyperparameter tuning for each OOD dataset.

Table 4.3: Ablation study of the proposed framework using CIFAR10 (ID) and TInr (OOD). While enforcing the structure hurts the ID accuracy slightly, it improves the OOD detection performance significantly. The remaining two combinations, (No, Yes, No) and (No, Yes, No), are not meaningful.

Union of 1D Subspaces	Orthogonal Subspaces	MC Samples	In Distribution Accuracy (%)	OOD AUROC
No	No	No	96.0	95.2
No	No	Yes	96.0	96.3
Yes	No	No	95.4	95.6
Yes	No	Yes	95.4	96.8
Yes	Yes	No	95.4	95.9
Yes	Yes	Yes	95.4	98.5

In the ablation study, Table 4.3 investigates the impact of enforcing structure on the OOD detection using spectral discrepancy. AUROC is computed by using spectral discrepancy for the different variants. This table shows that, while enforcing the proposed embedding slightly hurts the ID classification accuracy and does not improve the representation ability of the network, it is an effective technique to distinguish between ID and OOD samples. This table also shows the effect of MC samples, which are used to compute the probabilities. As expected, introducing MC sampling improves the OOD detection performance, regardless of the feature space structure. However, the improvement is more significant for networks on which our proposed structure is enforced. Further, MC sampling alone or enforcing 1D subspace alone does not make a significant difference. But the combination of 1D subspaces and MC samples improves the results significantly. This is mainly because our method is a probabilistic approach and only works in a probabilistic setting.

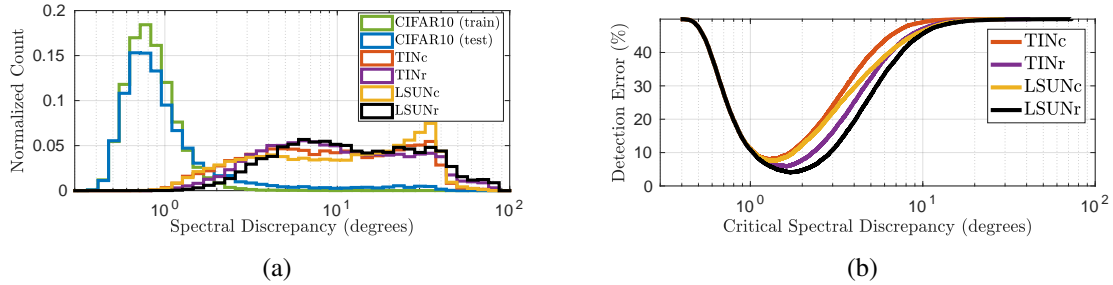


Figure 4.4: (a) Empirical probability distribution of the spectral discrepancy of samples belonging to CIFAR10 (ID) and different OOD datasets. (b) Detection error for different values of critical spectral discrepancy ϕ^* . Both the spectral discrepancy histogram and the best ϕ^* do not change significantly for different datasets.

As a guideline to set the value of the critical spectral discrepancy ϕ^* , Figure 4.4(a) shows the histogram of the spectral discrepancy for samples belonging to CIFAR10, as the ID dataset, and different real OOD datasets. It is evident that samples from both the testing and training set of the ID dataset follow a very similar behaviour. Thus, the training set can be used to estimate the possible interval of spectral discrepancies for the ID samples. For instance, about 98% of the

samples in CIFAR10 have a spectral discrepancy of less than 2 degrees. On the other hand, Figure 4.4(b) demonstrates the detection error for different values of the critical spectral discrepancy ϕ^* . This figure shows that best detection error is achieved by setting ϕ^* to a value in range $[1.3, 2]$ degrees, regardless of the OOD dataset. Hence, this figure shows that ϕ^* is not sensitive to the OOD dataset and can be set using only the training set. However, it should be mentioned that in general the best value for ϕ^* depends on the task at hand and the precision and/or recall requirements. As mentioned earlier, ϕ^* can also be set by many of the threshold estimation techniques such as [75]. More experimental results such as quantifying the impact of the number MC samples, robustness of the first singular vector to perturbations, and ROC curves are provided in Appendix B.

4.5 Conclusion

We showed that the distribution of the ID samples in the feature space plays an important role in the OOD detection. Particularly, we proposed to embed the ID samples into a low-dimensional feature space such that each known class lies on a 1-dimensional subspace. Such embedding gives us two main advantages in the OOD detection task: (i) ID samples occupy a tiny region in the space and (ii) ID samples have robust representatives. By exploiting these desirable features, our proposed method is able to outperform state-of-the-art methods in several performance metrics and different domains.

CHAPTER 5: ITERATIVE PROJECTION AND MATCHING: FINDING STRUCTURE-PRESERVING REPRESENTATIVES AND ITS APPLICATION TO COMPUTER VISION

Due to the proliferation of data gathering devices, an everincreasing amount of data is being generated and processed in different learning tasks. However, the ability to summarize and select good representatives from data is crucial in many applications. Furthermore, if a learning agent can achieve the same performance with fewer data, it is desirable to reduce the data size, as it will ease the requirements for data storage, communication, and processing. Thus, the goal of data selection is to capture the most structural information from a set of data. However, selecting a few representatives from a set of data points is not an easy task, as it might involve a combinatorial search over all the possible subsets. Thus, many different approximate solutions have been proposed in the literature. Approximate solutions has been proposed by exploiting either a convex [82, 83] or a sub-modular [84] cost function. More recently, authors in [2] and [85] proposed algorithms to select more representative samples, rather than focusing on diversity of the selected samples. These approaches have been shown to be effective in some computer vision tasks. However, since they rely on solving a convex optimization problem, their computational burden is not tractable for large data sets such as ImageNet. Furthermore, to solve such optimization problems, one need to set some hyperparameters, which is task- and dataset-dependant and oftentimes requires a grid search over all possible values¹.

This chapter presents a fast and accurate data selection method, in which the selected samples are optimized to span the subspace of all data. We propose a new selection algorithm, referred to as

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, M. Joneidi, N. Rahnavard, and M. Shah, “Iterative Projection and Matching: Finding Structure-preserving Representatives and Its Application to Computer Vision,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, 2019, © 2019 IEEE [55].

iterative projection and matching (IPM), with linear complexity w.r.t. the number of data, and without any parameter to be tuned ². In our algorithm, at each iteration, the maximum information from the structure of the data is captured by one selected sample, and the captured information is neglected in the next iterations by projection on the null-space of previously selected samples. Furthermore, the superiority of the proposed algorithm is shown on active learning for video action recognition dataset on UCF-101; learning using representatives on ImageNet; training a generative adversarial network (GAN) to generate multi-view images from a single-view input on CMU Multi-PIE dataset; and video summarization on UTE Egocentric dataset. In summary, this chapter makes the following contributions:

- The complexity of IPM is linear w.r.t. number of original data. Hence, IPM is tractable for larger datasets.
- IPM has no parameters for fine tuning, unlike some existing methods [2, 85]. This makes IPM dataset- and problem-independent.
- The superiority of the proposed algorithm is shown in different computer vision applications.

5.1 Iterative Projection and Matching (IPM)

Let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M \in \mathbb{R}^N$ be M given data points of dimension N . We define an $M \times N$ matrix, \mathbf{A} , such that \mathbf{a}_m^T is the m^{th} row of \mathbf{A} , for $m = 1, 2, \dots, M$. The goal is to reduce this matrix into a $K \times N$ matrix, \mathbf{A}_R , based on an optimality metric. Our proposed cost function for data selection is the error of projecting all the data onto the subspace spanned by the selected data. Mathematically,

²This study was done collaboratively with my colleague Mohsen Joneidi.

the optimization problem can be written as,

$$\underset{|\mathbb{T}|=K}{\operatorname{argmin}} \|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2. \quad (5.1)$$

$\pi_{\mathbb{T}}(\mathbf{A})$ is the projection of all the data on to the subspace spanned by the K rows of \mathbf{A} , indexed by \mathbb{T} . It is easy to show that $\pi_{\mathbb{T}}(\mathbf{A})$ can be expressed by a rank- K factorization, \mathbf{UV}^T , where $\mathbf{U} \in \mathbb{R}^{M \times K}$, $\mathbf{V}^T \in \mathbb{R}^{K \times N}$, and \mathbf{V}^T includes the K rows of \mathbf{A} , indexed by \mathbb{T} , and normalized to have unit length. Thus, our optimization problem can be rewritten as

$$\underset{\mathbf{U}, \mathbf{V}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{UV}^T\|_F^2 \text{ s.t. } \mathbf{v}_k \in \mathbb{A}, \quad (5.2)$$

where, $\mathbb{A} = \{\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \dots, \tilde{\mathbf{a}}_M\}$, $\tilde{\mathbf{a}}_m = \mathbf{a}_m / \|\mathbf{a}_m\|_2$, and \mathbf{v}_k is the k^{th} column of \mathbf{V} . To solve this problem in a tractable manner (linear time complexity with respect to M), we take a greedy approach and select only one sample at a time. In other words, we want to be able to represent \mathbf{A} as \mathbf{uv}^T , where $\mathbf{u} \in \mathbb{R}^M$, $\mathbf{v} \in \mathbb{R}^N$, and $\mathbf{v} \in \mathbb{A}$. The solution to this reduced optimization problem can be obtained efficiently by solving two consecutive problems as follows:

$$(\mathbf{u}, \mathbf{v}) = \underset{\mathbf{u}, \mathbf{v}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{uv}^T\|_F^2 \text{ s.t. } \|\mathbf{v}\| = 1, \quad (5.3a)$$

$$m^{(1)} = \underset{m}{\operatorname{argmax}} |\mathbf{v}^T \tilde{\mathbf{a}}_m|. \quad (5.3b)$$

$m^{(1)}$ is the index of the first selected representative. The first subproblem relaxes the original constraint $\mathbf{v}_k \in \mathbb{A}$ to $\|\mathbf{v}\| = 1$. This subproblem can be solved by setting \mathbf{v} as the first right singular

vector of \mathbf{A} . Time complexity order of computing the first singular vector of an $M \times N$ matrix is $O(MN)$ [73], i.e., linear time complexity. The second subproblem re-enforces the constraint by finding the closest sample in \mathbb{A} to the solution of the first subproblem. To select more samples, we first project the data matrix onto the null space of the selected sample and perform the same process till enough samples are collected. This means that the next sample will be selected by searching in the null space of the previous selected samples. This makes the overall time complexity of the proposed method $O(KNM)$. Furthermore, the sequential nature of our algorithm can be employed in applications such as active learning, where a new subset of data is added at each cycle. In the next section, we will investigate the effectiveness of IPM in active learning, as well as non-sequential scenarios such as video and dataset summarization.

To elaborate the steps in more details, Algorithm 2 demonstrates the proposed scheme in an algorithmic format. It is also worthwhile to mention that the condition that needs to be satisfied for a good performance is $K \leq N < M$. This ensures that the calculated singular vector is reliable and not impacted by noise. This condition is satisfied in subset selection scenarios, where the dataset is large, the number of selected samples is a lot less than the number of samples ($K \ll M$), and we have freedom over the dimension of the samples/features (N).

Algorithm 2 Iterative Projection and Matching Algorithm

Require: \mathbf{A} and K

Output: $\mathbf{A}_{\mathbb{T}}$

1: **Initialization:**

$\mathbf{A}^{(1)} \leftarrow \mathbf{A}$

$\mathbb{T} = \{\}$

for $k = 1, \dots, K$

2: $\mathbf{v} \leftarrow$ first right singular-vector of $\mathbf{A}^{(k)}$ by solving (5.3a)

3: $m^{(k)} \leftarrow$ index of the most correlated data with \mathbf{v} (5.3b)

4: $\mathbb{T} \leftarrow \mathbb{T} \cup m^{(k)}$

5: $\mathbf{A}^{(k+1)} \leftarrow \mathbf{A}^{(k)}(\mathbf{I} - \tilde{\mathbf{a}}_{m^{(k)}} \tilde{\mathbf{a}}_{m^{(k)}}^T)$ (null space projection)

end

Table 5.1: Classification accuracy (%) for action recognition on UCF-101, at different active learning cycles. The initial training set (cycle 1) is the same for all the methods. The accuracy for cycle 1 is 54.2% and the accuracy using the full training set (9537 samples) is 82.23%.

Mean samples/class	2	3	4	5	6	7	8
Random	60.1 \pm 0.7	65.1 \pm 1.2	68.2 \pm 1.7	69.9 \pm 1.4	71.7 \pm 0.6	73.0 \pm 0.6	74.8 \pm 0.5
Spectral Clustering	62.3 \pm 1.9	66.9 \pm 1.1	68.1 \pm 0.7	68.9 \pm 0.3	70.8 \pm 0.9	71.0 \pm 2.2	71.6 \pm 0.1
K-medoids	60.1 \pm 2.2	65.3 \pm 1.0	68.4 \pm 1.6	69.2 \pm 0.5	72.3 \pm 0.7	73.6 \pm 0.4	74.5 \pm 0.6
OMP	64.2 \pm 0.6	66.6 \pm 0.7	70.8 \pm 1.5	71.7 \pm 0.4	74.3 \pm 0.7	74.3 \pm 0.3	75.4 \pm 0.2
DS3 [2]	64.0 \pm 1.5	66.5 \pm 0.7	67.8 \pm 1.2	68.3 \pm 0.5	69.6 \pm 1.1	70.9 \pm 1.3	71.9 \pm 0.9
Uncertainty [86]	59.5 \pm 0.4	66.7 \pm 1.6	69.4 \pm 1.7	71.5 \pm 1.5	73.9 \pm 0.3	75.5 \pm 0.7	75.9 \pm 1.1
IPM	64.6 \pm 0.7	68.7 \pm 0.3	72.2 \pm 1.0	73.4 \pm 0.9	74.3 \pm 0.4	74.7 \pm 1.4	75.3 \pm 0.6
IPM + Uncertainty	64.3 \pm 0.4	69.4 \pm 0.8	72.8 \pm 1.0	73.8 \pm 0.9	76.2 \pm 1.0	76.3 \pm 0.3	77.9 \pm 0.2

5.2 Applications of IPM

To empirically demonstrate the behavior and effectiveness of the proposed selection technique, we have performed extensive sets of experiments considering several different scenarios. We divide our experiments into three different subsections. In Section 5.2.1, we use our algorithm in the *active learning* setting and show that IPM is able to reduce the labelling cost significantly, by selecting the most informative unlabeled samples. Next, in Section 5.2.2, we show the effectiveness of IPM in selecting the most informative representatives, by training the classifier using only a few representatives from each class. Lastly, in Section 5.2.3, the application of IPM for video summarization is exhibited.

5.2.1 Active Learning

Active learning aims at addressing the costly data labeling problem by iteratively training a model using a small number of labeled data, and then querying the labels of some selected data, using an acquisition function.

In active learning, the model is initially trained using a small set of labeled data (the initial training

set). Then, the acquisition function selects a few points from the pool of unlabeled data, asks an oracle (often a human expert) for the labels, and adds them to the training set. Next, a new model is trained on the updated training set. By repeating these steps, we can collect the *most informative* samples, which often result in significant reductions in the labeling cost. Now, the fundamental question in active learning is: Given a fixed labeling budget, what are the best unlabeled data instances to be selected for labeling for the best performance?

In many active learning frameworks, new data points are selected based on the model uncertainty. However, the effect of such selection only kicks in after the size of the training set is large enough, so we can have a reliable uncertainty measure. In this section, we show that the proposed selection method can effectively find the best representatives of the data and outperforms several recent uncertainty-based and algebraic selection methods.

In particular, we study IPM for active learning of video action recognition, using the 3D ResNet18 architecture, as described in [87]. The experiments are run on UCF-101 human action dataset [6], and the network is pretrained on Kinetics-400 dataset [88]. We provide the results on split 1.

To ensure that at least one sample per class exists in the training set, for the initial training, one sample per class is selected randomly and the fully-connected layer of the classifier is fine tuned. Then, at each active learning cycle, one sample per class is selected, without the knowledge of the labels, and added to the training set. Next, using the updated training set, the fully connected layer of the network is fine tuned for 60 epochs, using learning rate of 10^{-1} , weight decay of 10^{-3} , and batch size of 24 on 2 GPUs. Rest of the implementation and training settings are the same as [87]. Note that, in this experiment, fine-tuning is only performed to train the fully connected layer, because it achieved the best accuracy during the preliminary investigation for very small training sets, which is the scope of this experiment.

The selection is performed on the convolutional features extracted from the last convolutional

layer of the network. Table 5.1 shows the accuracy of the trained network at each active learning cycle for different selection methods. The high computational complexity of DS3 prevents its implementation on all the data [2]. So, we provide the results for DS3 only for the clustered version, meaning that one sample per cluster is selected using DS3 (clusters are obtained using spectral clustering). For spectral clustering results, the extracted features are clustered into 101 clusters, and one sample from each cluster is selected randomly. Furthermore, OMP, which stands for Orthogonal Matching Pursuit, selects the samples that are most correlated with the null space of the selected samples [89, 90].

The OMP approach is very sensitive to the outliers. Random outliers have low correlation with the samples and therefore a high correlation with the null space of the selected samples.

For uncertainty-based selection, Bayesian active learning [50, 86] is utilized. For that, a dropout unit with parameter 0.2 is added before the fully-connected layer and the uncertainty measure is computed by using 10 forward iterations (following the implementation in [50]). In our experiments, we use variation ratio³ as the uncertainty metric, which is shown to be the most reliable metric among several well-known metrics [86]. Also, for a fair comparison, the initial training set is the same for all the experiments at each run.

It is evident that, during the first few cycles, since the classifier is not able to generate reliable uncertainty score, uncertainty-based selection does not lead to a performance gain. In fact, random selection outperforms uncertainty-based selection. On the other hand, IPM is able to select the critical samples. In the first few active learning cycles, IPM is constantly outperforming other methods, which translates into significant reductions in labeling cost for applications such as video action recognition.

³Variation ratio of x is defined as $1 - \max_y p(y|x)$, which measures lack of confidence.

As the classifier is trained with more data, it is able to provide us with better uncertainty scores. Thus to enjoy the benefits of both IPM and uncertainty-based selection, we can use a compound selection criterion. For the extremely small datasets, samples should be selected only using IPM. However, as we collect more data, the uncertainty score should be integrated into the decision making process. Our proposed selection algorithm, unlike other methods, easily lends itself to such modification. At each selection iteration, instead of selecting the most correlated data with \mathbf{v} (line 3 in Algorithm 2), we can select the samples based on the following criterion:

$$m^* = \arg \max_m \alpha |\mathbf{v}^T \tilde{\mathbf{a}}_m| + (1 - \alpha) q(\mathbf{a}_m),$$

where $q(\cdot)$ is an uncertainty measure, e.g. variation ratios. Parameter α determines the relative importance of the IPM metric versus the uncertainty metric. To gradually increase the impact of $q(\cdot)$, as the model becomes more reliable, we start by setting $\alpha = 1$ and multiply it by decay rate of 0.95 at each active learning cycle. This compound selection criteria leads to better results for larger dataset sizes.

5.2.2 *Learning Using Representatives*

In this experiment, we consider the problem of learning using representatives. We find the best representatives for each class and use this reduced training set for learning. Finding representatives reduces the computation and storage requirements, and can even be used for tasks such as clustering. In the ideal case, if we collect the samples that contain enough information about the distribution of the whole dataset, the learning performance would be very close to the performance using all the data.

5.2.2.1 Representatives To Generate Multi-view Images Using GAN

Here, we present our experimental results on CMU Multi-PIE Face Database [91]. We use 249 subjects from the first session with 13 poses, 20 illuminations, and two expressions. Thus, there are $13 \times 20 \times 2$ images per subject. To investigate the effectiveness of the proposed selection, we use the selected samples to train a generative adversarial network (GAN) to generate multi-view images from a single-view input. For that, the GAN architecture proposed in [1] is employed. Following the experiment setup in [1], only 9 poses between $\frac{\pi}{6}$ and $\frac{5\pi}{6}$ are considered. Furthermore, the first 200 subjects are for training and the rest are for testing. Thus, the total size of the training set is 72,000,360 per subject. All the implementation details are same as [1], unless otherwise is stated.



Figure 5.1: Multi-view face generation results for a sample subject in testing set using CR-GAN [1]. The network is trained on reduced training set (9 images per subject) using random selection (first row), K-medoids (second row), DS3 [2] (third row), and IPM (fourth row). The fifth row shows the results generated by the network trained on all the data (360 images per subject). IPM-reduced dataset generates closest results to the complete dataset.

We select only 9 images from each subject (1800 total), and train the network with the reduced dataset for 300 epochs using the batch size of 36. Figure 5.1 shows the generated images of a subject in the testing set, using the trained network on the reduced dataset, as well as using the complete dataset. The network trained on samples selected by IPM (fourth row) is able to

generate more realistic images, with fewer artifacts, compared to other selection methods (rows 1-3). Furthermore, compared to the results using all the data (row 5), it is clear that IPM-reduced dataset generates the closest results to the complete dataset. This is because samples selected by IPM cover more angles of the subject, leading better training of the GAN. See Appendix C for further experiments and sample outputs.

Table 5.2: Identity dissimilarities between real and generated images by network trained on reduced (using different selection methods) and complete dataset.

Method	Random	K-Medoids	DS3	IPM
9 images / subject	0.5616	0.5993	0.6022	0.553
360 images / subject	0.5364			

For a quantitative performance investigation, we evaluate the identity similarities between the real and generated images. For that, we feed each pair of real and generated images to a ResNet18, trained on MS-Celeb-1M dataset [92], and obtain 256-dimensional features. ℓ_2 distances of features correspond to the face dissimilarity. Table 5.2 shows the normalized ℓ_2 distances between the real and generated images, averaged over all the images in the testing set. Our method outperforms other selection methods in this metric as well. Thus, from Figure 5.1 (qualitative) and Table 5.2 (quantitative), we can conclude that the IPM-reduced training set contains more information about the complete set, compared to other selection methods.

5.2.2.2 Finding Representatives for UCF-101 Dataset

Here, similar to Section 5.2.1, we use a 3D ResNet18 classifier pretrained on Kinetics-400 dataset, and the selection algorithms are performed on feature space generated by the output of the last convolutional layer. To find the representatives, we use the selection methods to sequentially find the most informative representatives from each class. After selecting the representatives, the fully

connected layer of the network is finetuned in the same manner as described in Section 5.2.1. Table 5.3 shows the performance of different selection methods for different numbers of representatives per class. As more samples are collected, the performance gap among different methods, including random, decreases. This is expected, since finding only one representative for each class is a much more difficult task, compared to choosing many, e.g. 6, representatives.

Table 5.3: Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.

Samples / Class	1	2	3	4	5	6
Random	54.6	64.7	69.2	70.5	72.9	74.0
K-medoids	61.0	67.7	69.4	70.9	71.7	72.0
OMP	51.1	64.6	70.7	72.8	73.0	74.5
DS3[2]	60.8	69.1	74.0	75.2	74.8	75.3
IPM	65.3	72.6	74.9	77.6	77.0	78.5

Using only one representative selected by IPM, we can achieve a classification accuracy of 65.3%, which is more than 10% improvement compared to random selection and more than 4% improvement compared to other competitors.

Figure 5.2 shows the t-SNE visualization [3] of the selection process for two randomly selected classes of UCF-101. To visualize the structure of the data, the contours represent the decision function of an SVM trained in this 2D space. Selection is performed on the original 512-dimensional feature space. This experiment illustrates that each IPM sample contains new structural information, as the selected samples are far away from each other in the t-SNE space, compared to other methods. Moreover, it is evident that as we collect more samples, the structure of the data is better captured by the samples selected by IPM, compared to other methods selecting the same number of representatives. The decision boundaries of the classifier trained on 5 IPM-selected samples look very similar to the boundaries learned from all the data. This leads to significant accuracy

improvements, as already discussed and exhibited in Table 5.3.

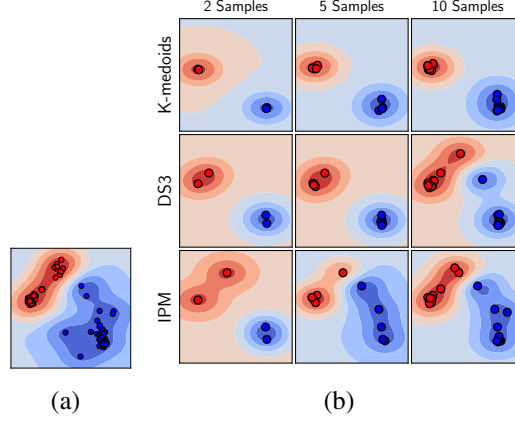


Figure 5.2: t-SNE visualization [3] of two randomly selected classes of UCF-101 dataset and their representatives selected by different methods. ((a)) Decision function learned by using all the data. The goal of selection is to preserve the structure with only a few representatives. ((b)) Decision function learned by using 2 (first column), 5 (second column), and 10 (third column) representatives per class, using K-medoids (first row), DS3 [2] (second row), and IPM (third row). IPM can capture the structure of the data better using the same number of selected samples.

5.2.2.3 Finding Representatives for ImageNet

In this section, we use ImageNet dataset [5] to show the effectiveness of IPM in selecting the representatives for image classification task. For that, first, we extract features from images in an unsupervised manner, using the method proposed in [93]. We then perform selection in the learned 128-dimensional space and perform k -nearest neighbors (k -NN) using the learned similarity metric, following the experiments in [93]. Here, we show that we can learn the feature space and the similarity metric in an unsupervised manner, as there is no shortage of unlabeled data, and use only a few labeled representatives to classify the data.

Due to the volume of this dataset, selection methods based on convex-relaxation, such as DS3 [2] and SMRS [85], fail to select class representatives in a tractable time. Table 5.4 shows the top-1

classification accuracy for the testing set using k -NN. Using less than 1% of the labels, we can achieve an accuracy of more than 25%, showing the potential benefits of the proposed approach for dataset reduction. Classification accuracy of k -NN, using the learned similarity metric, reflects the representativeness of the selected samples, thus highlighting the fact that IPM-selected samples preserve the structure of the data fairly well.

Table 5.4: Top-1 classification accuracy (%) on ImageNet, using selected representatives from each class. Accuracy using all the labeled data (1.2M samples) is 46.86%. Numbers in () show the size of the selected representatives as a % of the full training set.

Images per Class	1 (0.08%)	5 (0.4%)	10 (0.8%)	50 (4%)
Random	3.18	8.71	12.97	25.61
K-Medoids	11.78	17.01	17.56	26.86
IPM	12.50	21.69	25.26	30.77

5.2.3 Video Summarization

In this section, we evaluate the performance of the proposed selection algorithm on the video summarization task. The goal is to select key frames/clips and create a video summary, such that it contains the most essential contents of the video. We evaluate our approach on UT Egocentric (UTE) dataset [94, 95]. It contains 4 first-person videos of 3-5 hours of daily activities, recorded in an uncontrolled environment. Authors in [96] have provided text annotations for each 5-second segment of the video, as well as human-provided reference text summaries for each video. Following [96–98], the performance is evaluated in text domain. For that, a text summary is created by concatenating the text annotations associated with the selected clips. The generated summaries are compared with the reference summaries using the ROUGE metric [99]. As in prior work, we report f-measure and recall using the ROUGE-SU score with the same parameters as in [96–98].

Table 5.5: F-measure and recall scores using ROUGE-SU metric for UT Egocentric video summarization task. Results are reported for several supervised and unsupervised methods.

Method	F-measure	Recall
Selection Methods (Unsupervised)		
Random	26.30	23.73
Uniform	28.68	25.76
K-medoids	30.11	27.30
DS3	30.13	27.34
IPM	31.53	29.09
Supervised Summarization Methods		
SeqDPP [100]	28.87	26.83
Submod-V [98]	29.35	27.43
Submod-V+ [97]	34.15	31.59

Table 5.5 provides the results for two-minute-long summaries (24 5-second samples), generated by different methods. To generate results using K-medoids, DS3, and IPM, we use 1024-dimensional feature vectors extracted using GoogleNet [101], as described in [102]. Then, the features are clustered into 24 clusters using K-means and one sample is selected from each cluster using different selection techniques. The results are the mean results over all the 4 videos and over 100 runs. Furthermore, for the supervised methods, the results are as reported in [97]. *The proposed unsupervised selection method, IPM, is the closest competitor to the state-of-art supervised method proposed in [97], outperforming other unsupervised methods and some of the supervised methods.* These supervised methods split the dataset into training, and testing sets and use reference text or video summaries of the training set to learn to summarize the videos from the test set. This experiment demonstrates the strength of IPM and the potential benefits of employing it in more advanced unsupervised or supervised schemes.

5.3 Conclusions

A novel data selection algorithm, referred to as Iterative Projection and Matching (IPM) is presented, that selects the most informative data points in an iterative and greedy manner. We showed that our greedy approach, with linear complexity wrt the dataset size, is able to outperform state-of-the-art methods, which are based on convex relaxation, in several performance metrics such as projection error and running time. Furthermore, the effectiveness and compatibility of our approach are demonstrated in a wide array of applications such as active learning, video summarization, and learning from representatives. This motivates further investigation of the potential benefits and applications of IPM in other computer vision problems.

CHAPTER 6: FACE IMAGE RETRIEVAL WITH ATTRIBUTE MANIPULATION

The problem of image retrieval has been studied in many different applications such as product search [103, 104] and face recognition [105]. The standard problem formulation for image to image retrieval task is, given a query image, find the most similar images to the query image among all the images in the gallery. However, in many scenarios, it is necessary to *improve* and/or *adjust* the retrieval results by incorporating either the user’s feedback or by augmenting the query. This is due to the fact, in many cases, a perfect query image might not be readily available. Thus, it is desirable to give the user more control over the results. For example, in the context of fashion products, authors in [104, 106] exploit the user’s feedback to refine the search results iteratively. For instance, the method in [104] asks the user a series of visual multiple-choice questions to refine the search results and to eliminate the semantic gap between the user and the retrieval system. Another parallel approach is to augment the query with additional information, e.g., adjustment text, to modify the search results [7]. This is most often done by mapping the multi-modal query onto a joint embedding space [7, 107, 108]. These approaches treat different semantics the same and cannot prioritize a subset of attributes. Thus, the user is not able to define a customized distance metric and to assign importance to the attributes.

In this work, we introduce a new formulation for the image search task in the context of face image retrieval; and augment the query with both an adjustment vector and a preference vector. The **adjustment vector** is used to change the *presence* of certain attributes in the retrieved images, and the **preference vector** is used to assign the *importance* of the attribute in the results. To the best of our knowledge, this is the first work that can simultaneously adjust the attributes and assign preference values to them. Employing a preference vector gives the user the ability to customize

the similarity criteria. For instance, having eyeglasses might be more important to the user than having the same hair color. This criteria cannot be specified using only the adjustment vector, which is a limitation of existing retrieval methods. On the other hand, adjustment vector enables the user to use an imperfect query image for the search and adjust the attributes to achieve the ideal results. Furthermore, employing an adjustment vector, as opposed to an adjustment text, provides us with more flexibility, as many facial attributes cannot be easily described in text, for example different shades of brown hair.

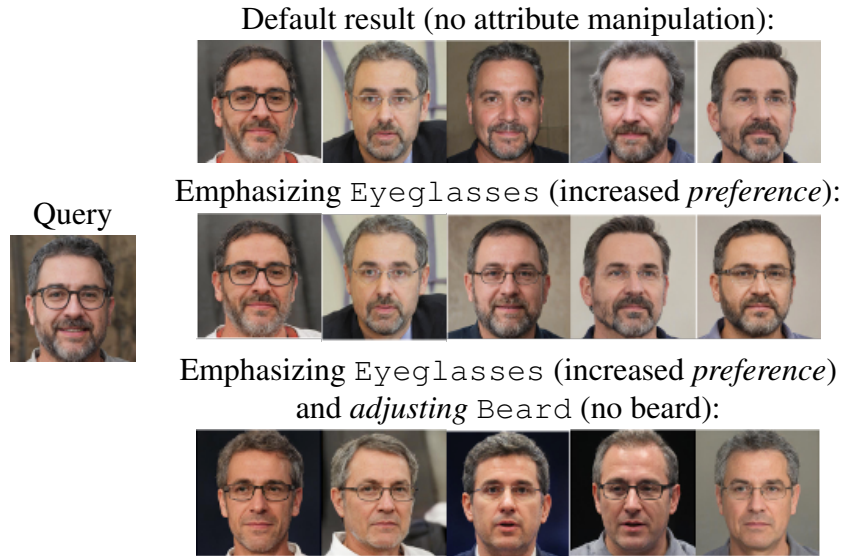


Figure 6.1: Example of face image retrieval by considering both the attribute *adjustment* and attribute *preference* specified by the user.

In the example provided in Figure 6.1, the impact of assigning a larger preference value and adjusting attributes are illustrated. In the middle row, the user has emphasized the attribute `Eyeglasses`, by assigning a larger *preference* value to it, which leads to all the top-5 retrieved images containing eyeglasses. The user can further fine-tune the results by *adjusting* any subset of the attributes. The bottom row shows the retrieved images after both emphasizing the attribute `Eyeglasses` and adjusting the attribute `Beard`, that's the beard has been removed and the eyeglasses are still

present.

To achieve this, we employ the recent advancements in generative adversarial networks (GANs). It has been shown that different semantic attributes are fairly disentangled in the latent space of StyleGAN [109, 110], even if the generator is trained in an unsupervised manner. This has been studied and experimentally verified in [8, 109]. This property provides us with an array of desirable features for face image retrieval. First, since the generator can be trained in an unsupervised manner, we do not need to have access to a lot of labeled data. A fairly small set of labelled data can be utilized to interpret the latent semantics learned by the generator. Second, the latent space provided by a well-trained StyleGAN provides us with an opportunity to both adjust the attributes and to assign preference to them. For that, we propose to obtain a set of disentangled attribute vectors in the latent space of StyleGAN. To disentangle the obtained attribute vectors, we enforce both *orthogonality* and *sparsity* constraints on them. We argue that, by making the attribute vectors sparse, we can decouple the entangled attributes even further. This is due to the fact that such attribute vectors can manipulate their corresponding semantic by affecting only a small subset of entries of the latent vector. This promotes selectivity among both the entries of the latent vector and the layers of the generator of the StyleGAN. On the other hand, by enforcing orthogonality, we can translate the dissimilarity between each image pair into dissimilarity between the attributes, assign preference to attributes, and define an attribute-weighted distance metric. In short, our contributions can be summarized as follows:

- We introduce a new face image retrieval framework that can simultaneously adjust the facial attributes and assign preference to different attributes in the retrieval task, employing the latent space of GANs (Section 6.2);
- We propose a new method to extract the directions of different attributes in the latent space, by learning all the attribute directions simultaneously and enforcing orthogonality and sparsity

constraints (Section 6.2.1);

- We utilize the learned attribute directions to define a weighted distance metric, to manipulate semantic attributes of the query, and to assign preference to different attributes for retrieval (Section 6.2.2); and
- The proposed method for image retrieval outperforms the recent state-of-the-art methods that use compositional learning or GANs for search (Section 6.3).

6.1 Related Work

Attribute-guided face image retrieval: There are many different approaches for image retrieval task based on metric learning such as [111–115], however they do not consider the task of retrieval with attribute manipulation. More similar to our attribute-guided retrieval setup, many of the methods utilize a query image and augment it with either an attribute adjustment text [7, 103, 107, 108, 116] or vector [104, 117]. Some of the prior work focuses on dialog-based interaction between the user and the retrieval agent, and improving the results in an iterative manner through user’s feedback [104, 106, 107]. Most of the attribute-aware retrieval methods *need huge amounts of labelled data* to generate a semantically meaningful latent space and distance metric [7, 103, 108, 116–118]. The method in [7] employs a new operation, referred to as residual gating, to create the joint embedding space between the image and text queries, which leads to state-of-the-art results among compositional learning methods such as [115, 119–123]. However, we propose to leverage the recent advancements in GAN architectures [21, 109, 110] and use the latent space generated by a GAN *trained in an unsupervised manner*, which significantly relaxes the requirements of access to labelled data. Furthermore, to the best of our knowledge, there has been no image retrieval method that can simultaneously *adjust* the attributes and assign *preference* to them.

Learning semantics in the latent space of GANs: Recent work have shown that the real image

data can be represented in the latent space of GANs, and specifically StyleGAN, with manifolds that have little curvature [8, 109, 124]. Such smooth behaviour can be enhanced by using loss functions [109, 125] or by modifying the generator architecture [110, 126]. A major benefit of the StyleGAN architecture [110] is the introduction of an intermediate latent space that does not need to follow any fixed sampling distribution, and the linear behaviour in this space is further enforced in [109] using path length regularization. It has been shown that this regularization leads to better Perceptual Path Length (PPL) score, which measures the perceptual score of the generated images after *linear* interpolation in the intermediate latent space. The authors in [8] employ this property and learn linear latent subspaces corresponding to different attributes. The authors in [8] proposed to orthogonalize the directions only during editing and in a sequential manner. This means that if the user wants to adjust multiple attributes, each new attribute direction is projected onto the null space of previous attributes. This approach has two main drawbacks. First, the final result depends on the order of applying the attribute adjustments. On the other hand, the sequential orthogonal projection makes it more difficult to define an attribute-guided distance metric and make the image retrieval very computationally expensive. In contrast, we propose to learn the latent subspaces simultaneously, and enforce orthogonality on the subspaces during the learning process. Furthermore, we study the impact of enforcing sparsity on disentangling the attributes.

6.2 Our Approach

Assume we have a set of M predefined facial attributes. In this setting, the query can be defined as a triplet $(\mathbf{x}_q, \mathbf{a}_q, \mathbf{p}_q)$, where \mathbf{x}_q is the query image, $\mathbf{a}_q \in [0, 1]^M$ is the vector specifying the intensity of each attribute (*attribute adjustment vector*), and $\mathbf{p}_q \in \mathbb{R}^{+M}$ is a vector containing positive real numbers indicating the *preference* for each attribute. The attribute adjustment vector (\mathbf{a}_q) can be used to *adjust* the search query. For instance, if the user assigns an intensity of 0 to

attribute *smiling*, the search results should not contain smiling faces, even though the query face is smiling. Also, the preference vector \mathbf{p}_q is independent of the adjustment vector \mathbf{a}_q , meaning that the value we assign as the preference value for each attribute does not depend on whether we are adjusting the attribute or not. The larger the preference value, the more similar the attribute should be to the query attribute. A preference value of 0 for a particular attribute means the user does not care about the presence/absence of that attribute. In this extreme case, the assigned attribute intensity will be ignored by the retrieval agent. The goal of our proposed framework is to rank the images in a gallery dataset based on the similarity with the query image, while considering both the *adjustments* and attribute *preferences* specified by the user.

To this end, we propose to perform the retrieval in the latent space of a StyleGAN [109]. This provides us with an array of desirable properties. First, as discussed in Section 6.1, it has been shown that different attributes can be manipulated fairly linearly in such a space [8, 109]. Second, using an unconditional StyleGAN gives us the opportunity to train it and its corresponding encoder using a large number of unlabeled data. We show how we can exploit a smaller number of labeled data to interpret the latent semantics learned by the StyleGAN.

The defining feature of StyleGAN architecture is the introduction of an intermediate latent vector, $\mathbf{w} \in \mathcal{W}$. In short, the generator of the StyleGAN consists of two main components: a mapping network and a synthesis network. The mapping network transforms the input latent vector to the intermediate latent space \mathcal{W} . Then the intermediate latent vector \mathbf{w} is used to modulate the convolution weights of the synthesis network, which generates the image.

It has also been shown that this intermediate latent space is consistently more disentangled than the input latent space, meaning that the attributes can be classified using a linear classifier more accurately in \mathcal{W} [109, 110]. This means that, given a binary attribute, there exists a hyperplane in \mathcal{W} that can separate the attribute classes. In other words, there exists a direction \mathbf{f} , i.e., the

direction orthogonal to the hyperplane, such that if we move the latent vector w along f , $w + \alpha f$, the class boundary can be crossed and the attribute can be turned to the opposite. α is a scalar which determines the displacement magnitude. Such directions can be obtained by training a linear classifier in \mathcal{W} , using labelled data. We argue that if we obtain an orthogonal and sparse basis set in \mathcal{W} , where each basis vector corresponds to a single attribute, we can easily adjust the attributes and define a weighted distance metric to retrieve images.

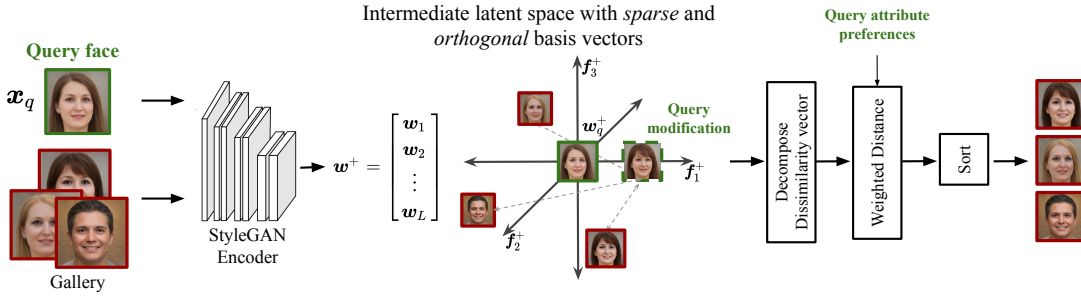


Figure 6.2: The overall architecture of the proposed face image retrieval framework. The intermediate latent space, \mathcal{W}^+ , is generated by employing StyleGAN encoder proposed in [4]. Then, the orthogonal and sparse basis vectors $\{f_m\}_{m=1}^M$ are extracted using a fairly small set of face images with attribute annotations. Utilizing the basis vectors, we adjust the query, decompose the dissimilarity vectors, and assign preference to different attributes.

The proposed retrieval framework can be summarized as follows. First, given a well-trained StyleGAN encoder trained on unlabeled data, a small set of labeled data (face images annotated with M attributes) are used to obtain an orthogonal basis set $\mathcal{F} = \{f_m\}_{m=1}^M$, $f_m \in \mathcal{W}, \forall m$, such that moving the latent vector along f_m only affects the m^{th} attribute (Section 6.2.1). Second, the obtained basis set \mathcal{F} is used to adjust the attributes, to define a weighted distance metric in \mathcal{W} , and to retrieve images (Section 6.2.2). The overall framework is shown Figure 6.2. Below, we discuss each of these two steps in more details.

6.2.1 Extracting Orthogonal Basis Set for Disentangled Semantics

As mentioned earlier, it has been empirically verified that different facial attributes can be manipulated fairly linearly in the latent space of StyleGAN [8, 109, 110, 124]. However, when there is more than one attribute, the obtained directions might be correlated with each other, meaning that adjusting one attribute using its corresponding direction might affect other attributes as well. To tackle this issue, let us examine how the intermediate latent vector is utilized to generate images. The latent vector is transformed to generate *styles* for each convolution layer in the synthesis network, using an affine transform, i.e., $\mathbf{s}_l = A_l(\mathbf{w})$. Here, \mathbf{s}_l stands for the style vector of l^{th} layer and $A_l(\cdot)$ is the learned affine transform of the pretrained StyleGAN. Each entry in \mathbf{s}_l is used to modulate the weights of a single convolution operator in the l^{th} layer. It has also been shown that instead of using a common latent vector \mathbf{w} for all the layers, we can extend the latent space and improve the encoding performance by finding a separate latent vector for each layer \mathbf{w}_l and producing the styles as $\mathbf{s}_l = A_l(\mathbf{w}_l)$. We refer to this space as the extended latent space \mathcal{W}^+ and represent the latent vector as the concatenation of layer-wise codes, $\mathbf{w}^+ = [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_L^T]^T \in \mathbb{R}^{d^+}$, and the attribute directions as $\mathbf{f}^+ \in \mathbb{R}^{d^+}$.

We argue that enforcing sparsity on the learned directions in \mathcal{W}^+ can effectively lead to disentangling the semantics and improved performance both for conditional image editing and the attribute-guided image retrieval. In other words, we look for attribute direction $\mathbf{f}^+ \in \mathcal{W}^+$ with minimum number of non-zero entries, while being able to classify the attributes accurately. This provides us with several advantages. First, it reduces the space of possible solutions and makes the learning problem more data-efficient. Thus, we are able to use a smaller set of labeled data to find the directions. Second, to manipulate the attribute in the latent space, $\mathbf{w}^+ + \alpha \mathbf{f}^+$, only a few entries of \mathbf{w}^+ are modified. Therefore, the learned direction \mathbf{f}^+ represents the minimum change necessary to manipulate the attribute. This leads to disentanglement of different attributes, as different attribute

directions only modify a very small, probably non-overlapping, subset of the entries. Finally, enforcing sparsity on the filters learned in extended latent space \mathcal{W}^+ also encourages non-uniform modification of the latent vectors across layers, as most of the entries are zeros. This is significant because the first few layers generate coarse details and later layers generate the finer details. Modifying a subset of layers means that the method is able to manipulate only the scales that are relevant to the attribute, leading to better disentanglement and accuracy.

Motivated by this, we propose to find an orthogonal and sparse basis set in the extended latent space, such that each basis vector corresponds to one of the attributes of interest. More specifically, given a set of N latent vectors $\{\mathbf{w}_n^+\}_{n=1}^N$ and their corresponding attribute labels $\{\mathbf{y}_n\}_{n=1}^N$, we look for $\mathcal{F} = \{\mathbf{f}_m^+\}_{m=1}^M$, $\mathbf{f}_m^+ \in \mathcal{W}^+$, such that $\mathbf{f}_m^{+T} \mathbf{f}_{m'}^+ = 0, m \neq m'$ and $\|\mathbf{f}_m^+\|_0 \leq \delta, \forall m$, where $\|\cdot\|_0$ is the ℓ_0 norm of a vector and indicates its number of nonzero entries. The sparsity condition can be enforced by regularizing the ℓ_1 norm of the attribute directions, which is the convex relaxation of the ℓ_0 norm. For our experiments, we employ 20,000 latent vectors ($N = 20,000$). Compared to many existing methods that use labelled data to create a semantically meaningful embedding, this is a large reduction in supervision requirements. For example, for quantitative comparisons with methods based on compositional learning in Section 6.3, their proposed models are trained with the full CelebA [127] training set, which contains about 160,000 faces.

To enforce the orthogonality constraint, at each iteration of learning the attribute vectors, we replace the learned set of attribute directions with its nearest orthogonal set. This problem is closely related to Procrustes problems, in which the goal is to find the closest orthonormal matrix to a given matrix [45]. Algorithm 3 summarizes the operations performed at each iteration on the learned attribute directions to find their nearest orthogonal set. In short, a matrix \mathbf{F} is created whose columns are the ℓ_2 -normalized version of learned directions. Then, the nearest *orthonormal* matrix to \mathbf{F} is calculated by finding the matrix $\hat{\mathbf{F}}$ that minimizes $\|\mathbf{F} - \hat{\mathbf{F}}\|_F^2$, such that $\hat{\mathbf{F}}^T \hat{\mathbf{F}} = \mathbf{I}$, where $\|\cdot\|_F$ denotes the Frobenius norm and \mathbf{I} is identity matrix. It can be shown that the solution to this prob-

lem is given by $\hat{\mathbf{F}} = \mathbf{F}(\mathbf{F}^T \mathbf{F})^{-\frac{1}{2}}$. Then, the columns of the orthonormal matrix $\hat{\mathbf{F}}$ are rescaled to have the same norms as \mathbf{F} .

Algorithm 3 Finding Nearest Orthogonal Set to a Set of Vectors.

Input: A set of vectors $\{\mathbf{f}_m\}_{m=1}^M$

- 1: $c_m = \|\mathbf{f}_m\|_2, \forall m$
 - 2: Create a matrix \mathbf{F} whose columns are $\mathbf{f}_1/c_1, \mathbf{f}_2/c_2, \dots, \mathbf{f}_M/c_M$
 - 3: Compute $\hat{\mathbf{F}} = \mathbf{F}(\mathbf{F}^T \mathbf{F})^{-\frac{1}{2}}$
 - 4: **return** $\{c_m \hat{\mathbf{f}}_m\}_{m=1}^M$, where $\hat{\mathbf{f}}_m$ is the m^{th} column of $\hat{\mathbf{F}}$
-

Algorithm 4 Extracting Orthogonal Basis Set for Disentangled Semantics

Input: Latent vectors $\{\mathbf{w}_n^+\}_{n=1}^N$ and their attribute labels $\{\mathbf{y}_n\}_{n=1}^N, \mathbf{y}_n \in \{0, 1\}^M$, classification loss function \mathcal{L}_c , regularization parameter λ , and a learning rate β

Output: A set of M orthogonal and sparse vectors, each corresponding to an attribute direction

- 1: Initialize the attribute directions $\{\mathbf{f}_m^+\}_{m=1}^M$ and biases b_m randomly
 - 2: **repeat**
 - 3: **for** each attribute $m = 1, \dots, M$ **do**
 - 4: Calculate $\hat{y}_{m,n} = \mathbf{f}_m^{+T} \mathbf{w}_n^+ + b_m$
 - 5: Compute Loss $\mathcal{L}_m = \sum_n \mathcal{L}_c(y_{m,n}, \hat{y}_{m,n}) + \lambda \|\mathbf{f}_m^+\|_1$
 - 6: $\mathbf{f}_m^+ = \mathbf{f}_m^+ - \beta \nabla_{\mathbf{f}} \mathcal{L}_m$
 - 7: $b_m = b_m - \beta \nabla_b \mathcal{L}_m$
 - 8: **end for**
 - 9: Replace $\{\mathbf{f}_m^+\}_{m=1}^M$ with its nearest orthogonal set using Alg 3
 - 10: **until** convergence
 - 11: Normalize $\mathbf{f}_m^+ = \mathbf{f}_m^+ / \|\mathbf{f}_m^+\|_2, \forall m$
 - 12: **return** $\{\mathbf{f}_m^+\}_{m=1}^M$
-

Algorithm 4 provides all the steps to extract the orthogonal sparse basis set in more details. At each iteration, after updating all the attribute directions using the gradient of the loss function, Algorithm 3 is used to enforce the orthogonality condition, by projecting the current iterate onto the feasible set (set of orthonormal matrices). In optimization literature, this feasible set is referred to as Stiefel manifold and the act of projection is referred to as retraction. It is shown that gradient descent with retraction onto Stiefel manifold converges to a critical point, under very mild conditions (see Theorem 2.5 in [128]). For our experiments, similar to prior research [8], we use hinge loss as the classification loss function \mathcal{L}_c .

6.2.2 Retrieval Using Orthogonal Decomposition

Dissimilarity decomposition and preference assignment: Given the obtained set of orthonormal directions, the query image \mathbf{w}_q^+ , and any other latent vector \mathbf{w}^+ , we decompose the dissimilarity vector $\mathbf{w}_q^+ - \mathbf{w}^+$ into its components. This can be done by projecting the dissimilarity vector onto each of the M attribute directions as:

$$\mathbf{d}_F = \mathbf{F}^T(\mathbf{w}_q^+ - \mathbf{w}^+) = \mathbf{F}^T \mathbf{w}_q^+ - \mathbf{F}^T \mathbf{w}^+, \quad (6.1)$$

where columns of $\mathbf{F} \in \mathbb{R}^{d^+ \times M}$ contains the M orthonormal vectors obtained by Algorithm 4. m^{th} entry of $\mathbf{d}_F \in \mathbb{R}^M$ represents the inner product of $\mathbf{w}_q^+ - \mathbf{w}^+$ with \mathbf{f}_m^+ . \mathbf{d}_F is the component of the dissimilarity vector that lies inside the subspace spanned by our M attribute directions. We can also compute the residual displacement that is not represented in this subspace as:

$$\mathbf{d}_I = (\mathbf{w}_q^+ - \mathbf{w}^+) - \mathcal{P}_F(\mathbf{w}_q^+ - \mathbf{w}^+) = (\mathbf{I} - \mathcal{P}_F)(\mathbf{w}_q^+ - \mathbf{w}^+), \quad (6.2)$$

where $\mathcal{P}_F = \mathbf{F}\mathbf{F}^T \in \mathbb{R}^{d^+ \times d^+}$ is the orthogonal projection matrix onto the subspace spanned by these vectors. This residual subspace contains information on the identity as well as other visual and semantic attributes not included in our M predefined facial attributes. Therefore, for a given query latent vector \mathbf{w}_q^+ and the attribute preference vector \mathbf{p}_q , we propose the following weighted distance metric from any other latent vector \mathbf{w}^+ as:

$$d(\mathbf{w}_q^+, \mathbf{w}^+, \mathbf{p}_q) = \mathbf{d}_F^T \mathbf{P} \mathbf{d}_F + \|\mathbf{d}_I\|_2^2, \quad (6.3)$$

where \mathbf{P} is an $M \times M$ diagonal matrix, whose diagonal entries contain the preference vector \mathbf{p}_q . The first term is the weighted Euclidean distance across different attribute directions (weighted attribute-aware distance), while the second term is the distance in the subspace not spanned by

these directions (attribute-independent distance). This gives the user the ability to fine-tune the contribution of each component to achieve the desired result. In the special case, where \mathbf{P} is set to identity matrix, this distance metric boils down to simple Euclidean distance in the latent space, $\|\mathbf{w}_q^+ - \mathbf{w}^+\|_2^2$.

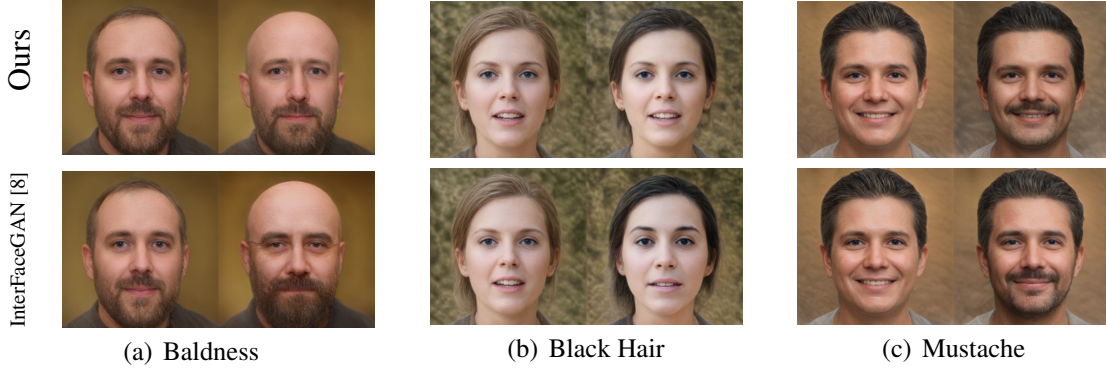


Figure 6.3: Qualitative evaluation of the learned attribute directions. In each pair of images, the image on the right is synthesized after moving the latent vector corresponding to the image on the left along an attribute direction. For attributes `Black Hair` and `Baldness`, the baseline is affecting the smile and the eyes as well, an artifact that is not present in the image manipulated by our method. For attribute `Mustache`, our method is able to add mustache to the face while not affecting the beard as much as the baseline.

Adjusting attributes: As mentioned earlier, we can adjust the m^{th} attribute in the query by moving its latent vector, \mathbf{w}_q^+ , along the direction corresponding to the m^{th} attribute, \mathbf{f}_m^+ , i.e., $\mathbf{w}_q^+ + \alpha \mathbf{f}_m^+$. Due to the definition of \mathbf{d}_I and \mathbf{d}_F , this operation will not affect \mathbf{d}_I , as it represents the displacement in the subspace not spanned by the attribute direction. Furthermore, such adjustment will only affect the m^{th} entry of \mathbf{d}_F . We can write \mathbf{d}_F for the adjusted latent vector as:

$$\mathbf{d}_F = \mathbf{F}^T(\mathbf{w}_q^+ + \alpha \mathbf{f}_m^+) - \mathbf{F}^T \mathbf{w}^+, \quad (6.4)$$

which, due to orthonormality, simply translates into adding α to the m^{th} entry of $\mathbf{F}^T \mathbf{w}_q^+$. Multiple attributes can be adjusted at the same time by modifying their corresponding entries independently.

Thus, we can manipulate the search results by updating \mathbf{d}_F as:

$$\mathbf{d}_F = T(\mathbf{a}_q, \mathbf{w}_q^+, \mathbf{F}) - \mathbf{F}^T \mathbf{w}^+,$$

where $\mathbf{a}_q \in [0, 1]^M$ is the attribute intensities provided by the user and $T(\cdot)$ is an affine transform that maps the range $[0, 1]$ to range of possible values for each entry of $\mathbf{F}^T \mathbf{w}_q^+$. The range of possible values, and therefore $T(\cdot)$, can be obtained using the training set. Specifically, the output of $T(\mathbf{a}_q, \mathbf{w}_q^+, \mathbf{F})$ is an M -dimensional vector, whose m^{th} entry is set as $a_{q,m}(a_{\max}^m - a_{\min}^m) + a_{\min}^m$, where a_{\max}^m and a_{\min}^m are the maximum and minimum value of $\mathbf{f}_m^{+T} \mathbf{w}_n$ over all the training feature vectors \mathbf{w}_n , respectively.

Implementation Details: We encode the face images in the training, query, and gallery sets using the StyleGAN encoder proposed in [4], trained in an unsupervised manner on FFHQ [110] dataset. This encoder is trained using the StyleGAN generator in order to be able to map real images onto the latent space, \mathcal{W}^+ . The latent vectors, $\{\mathbf{w}_n^+\}_{n=1}^N$, extracted from the training set are fed to Algorithm 4 to obtain the attribute directions $\{\mathbf{f}_m^+\}_{m=1}^M$. For latent vector of each query image, \mathbf{w}_q^+ , the dissimilarity vector is calculated by subtracting the query latent vector from each gallery latent vector. Using Equations (6.1) and (6.2), The dissimilarity vector, $\mathbf{w}_q^+ - \mathbf{w}^+$, is decomposed into \mathbf{d}_F and \mathbf{d}_I , which are then used to calculate the weighted distance (Equation (6.3)). This weighted distance metric is used to sort all the faces in gallery and retrieve the most similar images. The attributes can be adjusted either by moving the original latent vector, \mathbf{w}_q^+ along the corresponding attribute direct or, as shown in Equation (6.4), by modifying the projected latent vector.

Table 6.1: nDCG and identity similarity for different attribute-guided image retrieval methods, averaged over 1000 queries.

Number of retrieved images		5		10		20	
Method	Preference Assignment	nDCG	Identity Similarity	nDCG	Identity Similarity	nDCG	Identity Similarity
Attributes as Operators[120]	Not Applicable	0.730	0.824	0.720	0.823	0.711	0.824
TIRG [7]		0.794	0.847	0.781	0.844	0.776	0.840
Concat		0.804	0.841	0.806	0.838	0.805	0.822
Concat++		0.812	0.829	0.814	0.827	0.795	0.835
TIRG++ [7]		0.822	0.830	0.813	0.827	0.814	0.824
InterFaceGAN [8]	No Preference	0.568	0.838	0.570	0.835	0.571	0.832
	Identity Constrained	0.822	0.859	0.813	0.849	0.801	0.841
	Best nDCG	0.905	0.824	0.893	0.820	0.881	0.817
Ours	No Preference	0.595	0.849	0.586	0.845	0.583	0.841
	Identity Constrained	0.858	0.864	0.847	0.855	0.835	0.846
	Best nDCG	0.923	0.848	0.917	0.827	0.909	0.833

6.3 Experiments

In this section, we evaluate our proposed face image retrieval framework. We employ the StyleGAN architecture and the training details as discussed in [109]. For obtaining the attribute directions, generating queries, and creating the gallery set, CelebA dataset [127] is used. 20,000 samples, out of 160,000 from the training set are used for training the attribute directions, while the full test set, containing 19,962 faces, is used for creating queries and as the gallery data set. To the best of our knowledge, no other large-scale face dataset provides the ground truth for a large number of facial attributes. However, for qualitative results, we generate a much larger gallery set, containing 100,000 faces, by sampling from the latent space.

The search performance is quantified using two evaluation metrics. **Normalized discounted cumulative gain (nDCG)**, which measures the similarity of the query attributes, after making the adjustments specified by the user, with the search results, while giving more weight to the top results. nDCG is closely related to top- k accuracy for binary attributes, while giving the top results larger weight in a logarithmic manner (which makes it more suitable for ranking problems).

Furthermore, in contrast to top- k accuracy, nDCG can be used for real-valued attributes as well. **Identity Similarity** is calculated by embedding all the images onto the feature space generated by the Inception Resnet V1 architecture, as described in [129] and trained on VGGFace2 [130]. Then, the average cosine similarity between the embedded feature vector of the query face and the search results is used as a measure of identity similarity.

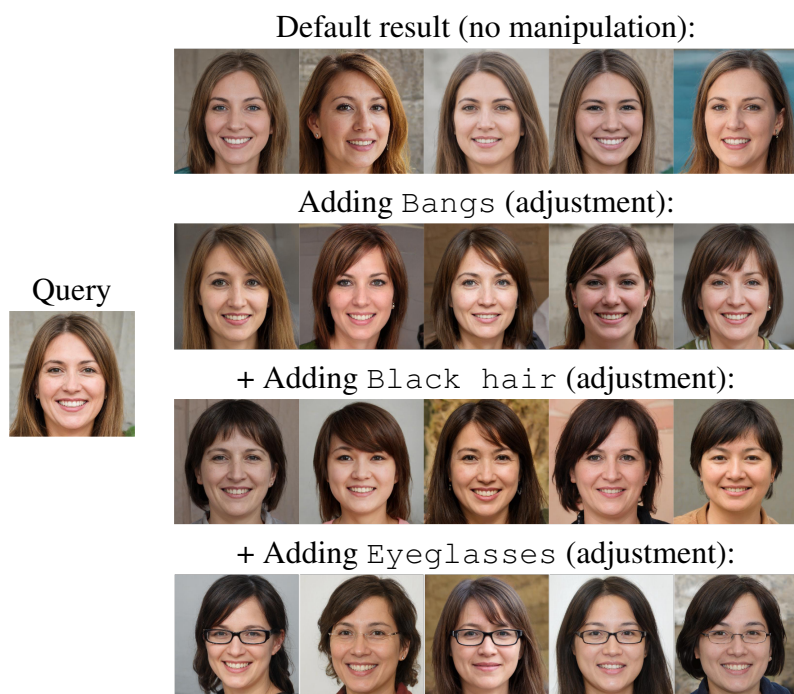
Unless otherwise stated, the regularization parameter λ and the learning rate β are set to 5×10^{-3} and 10^{-2} , respectively in Algorithm 4. λ is selected from the set $\{0, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$ by validating the obtained directions on the validation set of CelebA dataset. Best results for both the validation and test sets is achieved for $\lambda = 5 \times 10^{-3}$. The default value for attributes' preference is set to 1.

Qualitative Results: Figure 6.3 evaluates the obtained directions qualitatively for three attributes. In each pair of images, the left image is the starting point (image synthesized using a latent vector), and the image on the right shows the same image after adjusting a certain attribute (the image synthesized after moving the latent vector along the direction corresponding to the attribute). The top row illustrates the results obtained using the directions employing our proposed method and the middle row shows the results obtained by method in [8]. We argue that our proposed sparse attribute directions is able to preserve the identity better and also able to disentangle the attributes more accurately. For example, for attributes `Black Hair` and `Baldness`, the direction obtained by [8] is affecting the smile and shape of the eyes as well, an artifact that is not present in the image manipulated by our method. For attribute `Mustache`, our method is able to add mustache to the face while not affecting the beard as much as the baseline. This is due to the fact that, by enforcing sparsity, only the most relevant entries, and therefore layers, of the latent vector are modified.

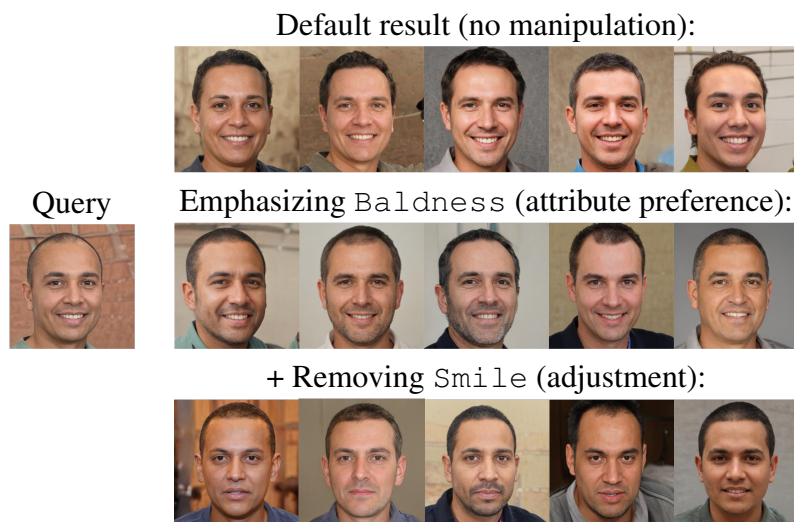
Figure 6.4 shows a few examples of retrieval results using the synthetic gallery set. It suggests that our retrieval approach performs well on different attributes for both adjusting and emphasizing

the attributes. It also shows that, our approach is able to adjust and emphasize multiple attributes at the same time, without affecting the other attributes much. For example, the last row in Figure 6.4(a) shows the results after adjusting three attributes, namely *Bangs*, *Black hair*, and *Eyeglasses*. Similarly, in the last row of Figure 6.4(b), the results are retrieved after adjusting attribute *Smile* and emphasizing attribute *Baldness*, by assigning a larger value to it.

Quantitative Results: Table 6.1 shows the nDCG and identity similarity for adjusting a single attribute using different attribute-guided image retrieval methods, averaged over 1000 queries. TIRG stands for Text Image Residual Gating, which uses text input to adjust the attributes [7]. We use the implementation provided by authors of [7, 120] to train the baseline models, using the full CelebA dataset. Similar to TIRG, *Concat* uses text queries and concatenates the feature vector extracted from the text input with feature extracted from the query image to perform the retrieval. TIRG++ and Concat++ stand for their improved versions, which does not use triplet loss, as discussed in detail in [7]. Unlike our proposed method, text inputs are not able to adjust the attributes in a continuous fashion and can only remove or add the attributes. Thus, for a fair comparison, we limit the attribute intensity vector provided to our framework to a binary vector, i.e., $\mathbf{a}_q \in \{0, 1\}^M$. However, our framework can also be used for continuous adjustment of attributes $\mathbf{a}_q \in [0, 1]^M$. Furthermore, the compositional learning methods cannot assign different preference values to different attributes. Thus, we evaluate the GAN-based methods under *four* different settings: (i) **Best nDCG:** This setting represents the case where the attribute preference for the changed attribute (not all the attributes) is set such that the best nDCG is achieved for each query. In this scenario, the nDCG of our method is significantly larger than the methods based on compositional learning, while the achieving the same identity similarity. (ii) **Identity constrained:** The attribute preference is set such that the identity similarity is at least as good as the best compositional learning method for each query. In this scenario, our proposed framework outperforms other competitors both in nDCG and identity similarity. As expected, the nDCG improvement is not as large as the



(a)



(b)

Figure 6.4: Qualitative evaluation of face image retrieval by considering both the *adjustment* and attribute *preference*. The user is able to both adjust multiple attributes in the query face and to customize the similarity metric by assigning preference to the attributes.

previous scenario. This shows that our method is able to preserve identities, while improving the attribute similarities. (iii) **No preference:** This setting corresponds to the scenario where user has no preference and all the attributes are treated the same. In this setting GAN-based methods underperform compositional learning methods in terms of nDCG. This shows the importance of assigning preference in the GAN-based methods. It is also worthwhile to mention that the compositional learning methods implicitly assign preference to the attribute being adjusted, as these models are trained using losses to adjust attributes. (iv) **Fixed attribute preference value:** In this setting, the preference value is the same for all the queries and does not depend on nDCG or identity similarity. Figure 6.5 illustrates the average top-5 nDCG and identity similarity for each value of attribute preference, averaged over all the queries. As expected, as we increase the preference for the target attribute, the attribute nDCG increases while the identity similarity decreases. However, even for the largest average nDCG, i.e., maximum attribute preference, the identity similarity is still comparable to the baselines in Table 6.1. This shows how the user can utilize the attribute preference to achieve the desired trade-off between identity and attribute retrieval. We want to stress out the fact the preference value is application-specific and cannot be optimized using the validation set.

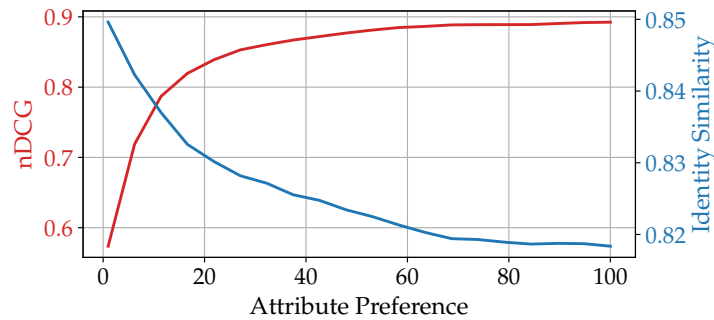


Figure 6.5: Impact of attribute preference on nDCG and identity similarity of the search results obtained by our method.

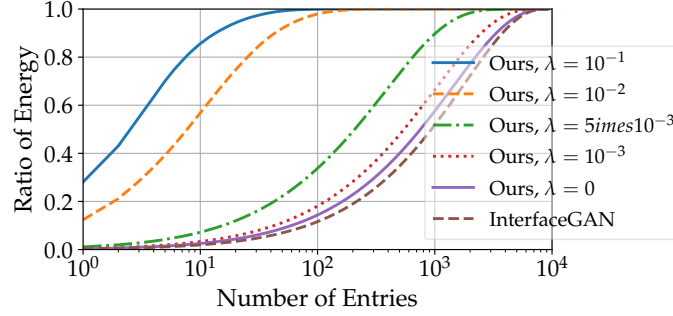


Figure 6.6: The energy concentrated in the top, most relevant, entries of the attribute vectors, averaged over all the attributes, for different values of the sparsity regularization parameter λ .

Table 6.2: Top-5 nDCG and identity similarity for different levels of sparsity, i.e., different values of the regularization parameter λ , averaged over 1000 queries.

Regularization parameter	nDCG	Identity Similarity
$\lambda = 0$ (no sparsity constraint)	0.826	0.863
$\lambda = 10^{-3}$	0.847	0.863
$\lambda = 5 \times 10^{-3}$	0.858	0.864
$\lambda = 10^{-2}$	0.849	0.866

Finally, to show the impact of sparsity on the selectivity of the attribute directions, Figure 6.6 illustrates the amount of energy in the most relevant entries of the attribute vectors for different values of sparsity regularization parameter λ , averaged over all the attributes. For instance, for the vectors trained using our method with $\lambda = 5 \times 10^{-3}$, about 1000 entries contain 95% the energy of the vector. This means that, in most cases, only 10% of the entries of a latent vector are modified to adjust the corresponding attribute. On the other hand, for vectors obtained using [8], the same amount energy is distributed over more than 5,000 entries. Table 6.2 shows the impact of sparsity on the image retrieval performance. It is clear that increasing λ up to 10^{-2} on the attribute direction can increase the attribute retrieval accuracy, in terms of nDCG, while keeping the identity similarity about the same. This shows that the sparse directions can successfully adjust the attribute, while

preserving the identity. The first row of the table, i.e., $\lambda = 0$ can also serve as an ablation study on the impact of enforcing only the orthogonality during the training. Comparing the results with the result of InterFaceGAN from Table 6.1, we can notice that, by only enforcing orthogonality, the same nDCG can be achieved with better identity similarity. It is worthwhile to mention that since our retrieval method depends on orthogonal decomposition of distances, we cannot report results without enforcing orthogonality. Additional implementation details and experiments, including more retrieval results on both CelebA and synthetic images, editing multiple attributes, and ablation study on number of training samples are provided in the supplementary materials.

6.4 Conclusion

In this chapter, a new setup for face image retrieval was proposed. The new setup considers a query face image, attribute modifiers, and attribute preference as input constraints to retrieve the most compatible face from a gallery set. While the attribute modifiers define which attributes to manipulate in the query image, the attribute preference set the importance or weight assigned to each attribute when compared to a gallery image. We proposed a model that leverages the StyleGAN latent space characteristics to learn sparse and orthogonal attribute directions to increase control over each attribute dimension and to allow adjusting multiple attributes at the same time, while reducing unwanted changes in the rest of the attributes. The proposed setup was evaluated on CelebA and compared to a set of state-of-the-art baselines showing better retrieval performance.

CHAPTER 7: ADAPTIVE NON-UNIFORM COMPRESSIVE SAMPLING FOR TIME-VARYING SIGNALS

The goal of the compressed sensing (CS) problem is to recover the signal $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ with length N from its undersampled random projections, also referred to as measurements. M random projections are generated using a measurement matrix $\Phi \in \mathbb{R}^{M \times N}$ from the linear measurement process, $\mathbf{y} = \Phi \mathbf{x} + \mathbf{n}$, where $\mathbf{y} = [y_1, y_2, \dots, y_M]^T$ represents the measurement vector and \mathbf{n} denotes the corrupting noise¹.

Signal \mathbf{x} is said to be K -sparse if it has at most K non-zero entries in a proper basis. The sparsity of \mathbf{x} can be exploited to find a unique solution of the underdetermined system equation with high probability from $\mathcal{O}(K \log(\frac{N}{K}))$ measurements [23].

In this chapter, we consider the problem of reconstructing a correlated time series of such compressible vectors from their noisy undersampled measurement. Particularly, we are interested in approximating the time series $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots\}$ from the measurement time series $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots\}$. In many real-world applications, the signal of interest has a substantial correlation in time. The main idea is to incorporate the knowledge from the previous estimates of the signal to achieve a more accurate estimation of the signal at the current time step.

Moreover, in many applications, different parts of the signal have different recovery requirements. Thus, different coefficients of the signal have different *importance levels*. For instance, in video processing, it is desired to recover the salient area more accurately. Moreover, if the signal is sparse in canonical basis, we are interested in reconstructing the large coefficients with less error.

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, M. Joneidi, and N. Rahnavard, “Adaptive non-uniform compressive sampling for time-varying signals,” in *51st Annual Conference on Information Sciences and Systems, CISS 2017*, 2017, © 2017 IEEE [131].

Non-uniform acquisition and recovery of signal is desirable in many applications such as image processing [132], camera sensor networks [133, 134], wireless sensor networks [135], collaborative vector estimation [136], component analysis [137, 138], and internet of things [139].

In this work, we propose an adaptive framework to design a non-uniform measurement matrix, which contrast with *dynamic CS algorithms* [140–142] focusing only on the recovery step. Our method is also distinct from the *adaptive CS* [143–145] methods that are concerned with reconstructing signals, which are static over time. Here, similar to adaptive CS, the main idea is to concentrate the sensing energy on the more important coefficients, by designing a proper measurement matrix. However, due to dynamic nature of the problem, the algorithm should not make firm decisions about the location of more important coefficients. Hence, soft importance level information is advantageous. To infer the importance level of each coefficient at each time step, a generative model is imposed on the coefficients and the parameters of the model are updated in an online fashion.

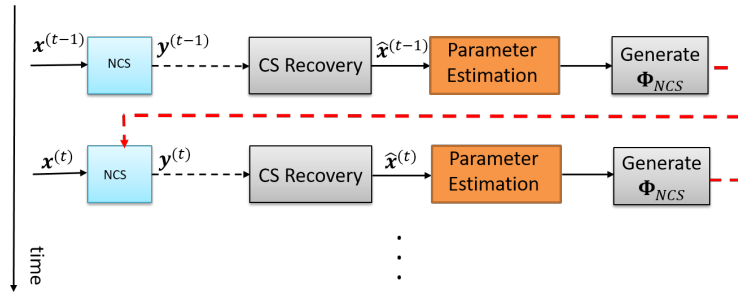


Figure 7.1: Overall block diagram of the proposed framework. Reconstructed signal, at each time step, is utilized to generate the measurement matrix.

Figure 7.1 shows the overall architecture of the proposed method. At each time step, after reconstructing the signal, by using a conventional CS recovery algorithm, the importance levels of the coefficients are inferred mathematically. The importance levels are further employed to design the measurement matrix for the next time step.

The rest of this chapter is organized as follows. In Section 7.1, the system model is presented. Then, the generative model of the proposed Bayesian framework is introduced in Section 7.2. In Section 7.3, the inferred importance level information are used to design the measurement matrix for sensing. Finally, Section 7.4 presents the simulation results and Section 7.5 draws conclusions.

7.1 System Model

We consider recovery of a vector-valued time series $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots\}$ from the linear measurements given by

$$\mathbf{y}^{(t)} = \Phi^{(t)} \mathbf{x}^{(t)} + \mathbf{n}^{(t)}, \quad t = 1, 2, \dots \quad (7.1)$$

where $\mathbf{n}^{(t)} \in \mathbb{R}^M$ represents the noise and is modeled as an additive white Gaussian noise (AWGN) with $\mathbf{n}^{(t)} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_M)$.

It is assumed that the signal of interest $\mathbf{x}^{(t)}$ is compressible and contains coefficients with different importance levels, which are not known a priori. In many scenarios, it is desirable to have non-uniform recovery performance on different parts of signal. More important coefficients may correspond to support of a sparse vector or the salient area in a video frame.

We also assume that, at each time step, using the estimation of the signal $\hat{\mathbf{x}}^{(t)}$, more important coefficients are tagged using a possibly erroneous algorithm. The variable $\alpha^{(t)}$ marks the detected *region of interest* (ROI) in the signal at time t . Specifically, $\alpha_n^{(t)} = 1$, if the n^{th} coefficient of the signal is detected to be in the ROI, and $\alpha_n^{(t)} = 0$ otherwise. However, due to sensing failure, error in recovery of $\hat{\mathbf{x}}^{(t)}$, and/or misdetection of the ROI, $\alpha^{(t)}$ may contain erroneous elements.

As mentioned earlier, the signal of interest often exhibits substantial temporal correlation. Here, we assume that ROI, and therefore the support of non-zero entries in $\alpha^{(t)}$, changes slowly in time.

Our goal is to employ the temporal correlation to infer reliable importance level information and employ the importance levels to design a non-uniform measurement matrix.

7.2 Bayesian Inference of Importance Levels

To extract reliable information from possibly faulty ROI data $\alpha^{(t)}$, we propose to employ Bayesian inference. In Bayesian framework, the goal is to infer the probability distribution of hidden variables given the observations. The hidden variables are often the parameters that are desired to be estimated. Specifically, in our model, the following hidden variables are introduced:

1. Coefficient-specific reliability $u_n \in \{0, 1\}$, which is either 0 or 1 and describes the reliability of ROI data of the n^{th} coefficient.
2. Overall reliability $r \in [0, 1]$, denoting the overall trustworthiness of the ROI detection algorithm. For small values of r , the algorithm is more prone to reporting faulty data. A generally reliable algorithm will report trustworthy measurements on most of the coefficients.
3. Importance level for each coefficient $c_n \in [0, 1]$, describing the probability that coefficient n is in ROI.

As mentioned earlier, in the proposed generative model, α_n is the observed variable. If $\alpha_n = 1$, the n^{th} coefficient is detected to be in ROI, and $\alpha_n = 0$ otherwise. In this model, coefficient-specific reliability and overall reliability model the faulty data. Without them, all the observations would be assumed to be trustworthy, which is not the case in real-world scenarios.

Figure 7.2 illustrates the graphical representation of the proposed generative model. The arrows in the graph represent the dependency among the variables. Hence, the observed ROI data depends on the actual importance level of the coefficients and the reliability of algorithm in detecting the

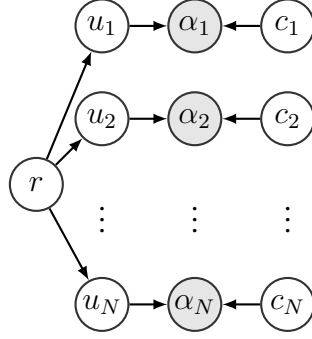


Figure 7.2: Graphical representation of the generative model.

ROI coefficients. The goal of the inference algorithm is to obtain the probability distribution of the overall reliability, coefficient-specific reliability, and the importance levels, given the ROI data. At each time step t , the proposed model can be formulated as follows, for $n = 1, \dots, N$:

$$\begin{aligned}
 r &\sim \text{Beta}(b^1, b^0) \\
 u_n &\sim \text{Bernoulli}(r) \\
 c_n &\sim \text{Beta}(\beta_n^1, \beta_n^0) \\
 \alpha_n^{(t)} &\sim u_n \text{Bernoulli}(c_n) + (1 - u_n) \text{Bernoulli}(1 - c_n)
 \end{aligned} \tag{7.2}$$

The observed variable $\alpha_n^{(t)}$ is modeled with summation of two Bernoulli distributions. This means that if the ROI data for n^{th} coefficient is reliable, i.e. $u_n = 1$, α_n will be sampled from a Bernoulli distribution with true parameter for importance level, i.e. c_n . Otherwise, it will be sampled from $\text{Bernoulli}(1 - c_n)$ and will be more probable to report faulty data. Since c_n is used as the parameter of a Bernoulli distribution, it is the natural choice to model it with a Beta distribution. This is due to the fact that the conjugate prior for Bernoulli distribution is Beta distribution.

Similarly, the variable representing the overall reliability, i.e. r , is modeled with a Beta distribution. This is because the coefficient-specific reliability variables are sampled from $\text{Bernoulli}(r)$. This

means that if the ROI detection is reliable in general, ROI data on most of the coefficients will be reliable. This prior links the performance of the algorithm on different coefficients and reduces the chance of overfitting the coefficient-specific reliability.

As mentioned earlier, the goal of the inference algorithm is to obtain the distribution of hidden variables, given the observations, i.e. $\mathbb{P}\{\mathbf{c}, \mathbf{u}, r | \mathcal{A}\}$. For compactness of notation, we set $\mathbf{c} = \{c_1, c_2, \dots, c_N\}$, $\mathbf{u} = \{u_1, u_2, \dots, u_N\}$, and $\mathcal{A} = \{\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \dots\}$. At each time step, after receiving the ROI data, the distribution of hidden variables are inferred by exploiting the data and the prior belief, represented by the prior distribution $\mathbb{P}\{\mathbf{c}, \mathbf{u}, r\}$.

For that, we need to specify the joint distribution of the observation and the hidden variables. Specifically, using the model formulated in (7.2), we have:

$$\mathbb{P}\{\mathcal{A}, \mathbf{u}, \mathbf{c}, r\} = \prod_{t=1}^{\infty} \prod_{n=1}^N \mathbb{P}\{\alpha_n^{(t)} | u_n, c_n\} \mathbb{P}\{u_n | r\} \mathbb{P}\{c_n | \beta_n^1, \beta_n^0\} \mathbb{P}\{r | b^1, b^0\} \quad (7.3)$$

However, due to obvious practical reasons and to limit the history of the inference, the inference is performed using a few of recent observations. For that, a sliding window of length W is utilized and the parameters of the posterior distributions are inferred using only the last W observations.

To infer the importance level of the coefficients as well as the reliability of the ROI data, we need to find the posterior distribution given the ROI data, i.e., $\mathbb{P}\{\mathbf{u}, \mathbf{c}, r | \mathcal{A}\}$. However, directly obtaining the posterior distributions is not computationally feasible and results in explosive number of probability factors growing exponentially with number of coefficients. To handle the intractable integrals of the inference procedure, *variational inference* is often employed [146–148].

In variational inference, the posterior distribution is assumed to be fully factorized over all the hidden variables. In other words, the posterior distribution is being approximated by a family of

distributions, for which the inference procedure is tractable. For our model, the fully factorized approximation of the posterior distribution, also referred to as the variational distribution, is defined as:

$$\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\} = \prod_n \mathbb{Q}\{c_n | \hat{\beta}_n^1, \hat{\beta}_n^0\} \mathbb{Q}\{u_n | \tau_n\} \mathbb{Q}\{r_n | \hat{b}^1, \hat{b}^0\}. \quad (7.4)$$

where \hat{b}^1 , \hat{b}^0 , $\hat{\beta}_n^1$, $\hat{\beta}_n^0$, and τ_n are the parameters of the factorized distributions. By introducing the variable τ_n , we are seeking the best approximate of $\mathbb{P}\{\mathbf{u}, \mathbf{c}, r | \mathcal{A}\}$ among all the distributions $\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\}$, by factorizing the distribution over *disjoint* groups of hidden variables. \mathbf{u} , \mathbf{c} , and r . It is worthwhile to mention that we make no further assumption about the distributions and their functional forms.

Specifically, we aim to find the best set of distributions and parameters that maximizes the lower bound of log likelihood of the observations [147, 149]. The lower bound of log-likelihood of the observations can be written as [149, Chapter 10]:

$$\begin{aligned} \ln(\mathbb{P}\{\mathcal{A}\}) &\geq \int \mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\} \ln(\mathbb{P}\{\mathcal{A}, \mathbf{c}, \mathbf{u}, r\}) - \int \mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\} \ln(\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\}) \\ &= \mathbb{E}\{\ln(\mathbb{P}\{\mathcal{A}, \mathbf{c}, \mathbf{u}, r\})\} - \mathbb{E}\{\ln(\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\})\} \triangleq \mathcal{L}(\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\}), \end{aligned} \quad (7.5)$$

where the expected value is with respect to variational distribution. Hence, the problem boils down to maximizing $\mathcal{L}(\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\})$ to find the best variational distributions. Since the lower bound is concave with respect to each of the factorized distributions, i.e., $\mathbb{Q}\{c_n | \hat{\beta}_n^1, \hat{\beta}_n^0\}$, $\mathbb{Q}\{u_n | \tau_n\}$, and $\mathbb{Q}\{r_n | \hat{b}^1, \hat{b}^0\}$, we can determine the best approximate distributions by maximizing $\mathcal{L}(\mathbb{Q}\{\mathbf{c}, \mathbf{u}, r\})$ with respect to one factor at a time [149]. Thus, at each step, the lower bound is maximized over one factor, keeping all the other distributions. This procedure is repeated until convergence.

For simplicity of notation, let us denote the whole set of hidden variables with $\mathbf{Z} = \{\{c_n\}, \{u_n\}, r\}$. In (7.4), \mathbf{Z} is divided into disjoint groups $Z_i, i = 1, \dots$, where each Z_i is representing one of the hidden variables in \mathbf{Z} . By maximizing the lower bound $\mathcal{L}(\mathbb{Q}\{\mathbf{Z}\})$, the variational distribution of

each partition $\mathbb{Q}\{\mathbf{Z}_i\}$ is given by [149, Chapter 10]:

$$\ln(\mathbb{Q}\{\mathbf{Z}_i\}) = \mathbb{E}_{j \neq i} \{\ln(\mathbb{P}\{\mathcal{A}, \mathbf{Z}\})\} + \text{const}, \quad (7.6)$$

where $\mathbb{E}_{j \neq i} \{\cdot\}$ is the expectation with respect to distributions $\mathbb{Q}\{\mathbf{Z}_j\}, j \neq i$. Then by plugging in $\mathbb{P}\{\mathcal{A}, \mathbf{Z}\} = \mathbb{P}\{\mathcal{A}, \mathbf{c}, \mathbf{u}, r\}$ from (7.3) and employing the exponential form of the distributions, the variational distributions can be obtained. The constant value is determined by normalizing the distribution.

Using (7.6), we can derive closed form expressions for parameters of the variational distributions. At each time step, after receiving the new observation vector, $\boldsymbol{\alpha}^{(t)}$, the distribution of the hidden variables are updated using the derived update rules. Then, the updated distributions are used to concentrate the sensing energy on the more important coefficients of the signal.

7.3 Measurement Matrix Design

In this section, the distributions of the importance levels are exploited to design the measurement matrix at each time step $\Phi^{(t)}$. The idea is to employ the information extracted from the previous measurements and focus the sensing energy on the ROI coefficients.

In conventional compressive sensing methods, the sensing energy is distributed uniformly among the coefficients of the signal. In many standard methods, it is assumed that the column of the measurement matrix are scaled to be of unit norm. Thus, the total amount of sensing energy is $\|\Phi\|_F^2 = N$. In this work, we also assume that the available sensing energy is N . A constraint on the available sensing energy is necessary for any practical implementation. Also, without the constraint, the issue of noise would be irrelevant.

In adaptive sensing procedures [143–145, 150], no energy is allocated to the coefficients that are not likely to be in support of the signal, i.e., ROI. However, in our problem, since we are dealing with time-varying signals, such hard decisions should be avoided.

The key aspect of the proposed method is the allocation of sensing energy across the coefficients of the signal. In Section 7.2, a Bayesian framework is introduced to obtain the distribution of the importance of each coefficient. Specifically, the norm of the n^{th} column of the measurement matrix $\Phi^{(t)}$ is given as:

$$\gamma_n^{(t)} = \sqrt{N} \frac{\bar{c}_n}{\eta} \quad (7.7)$$

where \bar{c}_n is the expected value of the importance level of the n^{th} coefficient of the signal, i.e. $\bar{c}_n = \mathbb{E}_{\mathbb{Q}\{c_n\}}\{c_n\}$. and η is a constant to ensure that the energy constraint is met. Specifically, for $\eta = \sqrt{\sum_n \bar{c}_n^2}$, we will have $\|\Phi\|_F^2 = N$.

Thus, at each time step the estimate of the signal is used to update the distribution of the hidden variables. Then, the inferred importance levels are exploited to tune the energy allocated to each coefficient of the signal.

7.4 Numerical Experiments

In this section, a series of numerical experiments are presented to highlight the performance gain of the ANCS. The primary performance metric used in our studies is time averaged normalized MSE (TNMSE), which is defined as $\frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{x}^{(t)} - \hat{\mathbf{x}}^{(t)}\|_2^2}{\|\mathbf{x}^{(t)}\|_2^2}$. where T is the number of time slots of the signal, $\|\cdot\|_2$ is the ℓ_2 -norm of a vector, and $\hat{\mathbf{x}}^{(t)}$ is the estimate of $\mathbf{x}^{(t)}$ at time t .

The parameters of the algorithm are set as follows. Since, no prior information is assumed on the importance levels of the coefficients, the parameters are initialized as $\beta_n^1 = 1 = \beta_n^0 = 1, \forall n$. This

choice of parameters results in a uniform distribution for the importance levels. To initialize b^1 and b^0 , it is reasonable to assume that at least half of the measurements are reliable. In our numerical experiments, we initialized $b^1 = 3$ and $b^0 = 1$, which means on average 75% of the measurements are trustworthy. The maximum number of iterations for the inference algorithm is set to 40, with possibility of early termination if $\frac{\sum_n (\bar{c}_n^{(k)} - \bar{c}_n^{(k-1)})^2}{\sum_n (\bar{c}_n^{(k-1)})^2} \leq 10^{-6}$ at k^{th} iteration. Moreover, a window length of $W = 5$ is used.

In all the simulations to construct the measurement matrices, elements of the matrix were drawn from an i.i.d zero mean Gaussian distribution. For uniform sampling, the columns of the matrices are scaled to have unit norm. On the other hand, for ANCS, (7.7) is used to realize the non-uniform distribution of energy among the columns. The total sensing energy of all the methods is assumed to be the same, i.e. $\|\Phi^{(t)}\|_F^2 = N$, $\forall t$.

As a performance benchmark and to quantify the performance improvement obtained by the ANCS, we exploit the proposed method as the sampling step of an ℓ_1 minimization recovery algorithm. Specifically, the estimate of the signal is obtained by solving an ℓ_1 minimization problem, given by:

$$\hat{\mathbf{x}}^{(t)} = \arg \min \|\mathbf{x}\|_1, \text{ s.t. } \|\mathbf{y}^{(t)} - \Phi^{(t)}\mathbf{x}\|_2 \leq c,$$

where $\|\mathbf{x}\|_1 = \sum_n |x_n|$ and c is set to be equal to $\sigma_n \sqrt{M}$. To solve the problem, CVX [151, 152], which is a toolbox for specifying and solving convex problems, is used.

7.4.1 Performance evaluation for sparse signals in canonical basis

For the first experiment, the performance gain of ANCS is quantified for the signals that are sparse in canonical basis. To model the temporal correlation, both in amplitude and support of the signal, the signal is assumed to be outcome of two random processes. Specifically, a binary vector $\mathbf{s}^{(t)} =$

$[s_1^{(t)}, \dots, s_N^{(t)}]^T$ describes the support of the signal at time t . $s_n = 1$ indicates the coefficients in the support and $s_n^{(t)} = 0$ denotes the zero coefficients. Coefficients of $\mathbf{s}^{(t)}$ are assumed to be independent and a Markov chain process is defined for each of the coefficients. The Markov chain processes are described by $p_{01} = \mathbb{P}\{s_n^{(t)} = 1 | s_n^{(t-1)} = 0\}$ and $\lambda = \mathbb{P}\{s_n^{(t)} = 1\}$, $\forall n, t$. Thus, λ is related to the sparsity level of signal.

Furthermore, a second process models the amplitude of the large coefficients. We employ an independent Gauss-Markov process for each of the coefficients of the signal. Amplitude of the n^{th} coefficient evolves over time as: $a_n^{(t)} = (1 - \rho)a_n^{(t-1)} + \rho\nu_n^{(t)}$. Here, ρ is a constant between 0 and 1 and controls the degree of correlation. For $\rho = 1$, the amplitude would be an uncorrelated Gaussian random process. $\nu_n^{(t)}$ is the amount of variation among two consecutive time steps and is modeled with $\mathcal{N}(0, \sigma_L^2)$. Thus the mean of the process is assumed to be 0. At each time step, the coefficients of the signal are constructed as $x_n^{(t)} = a_n^{(t)} s_n^{(t)}$.

The simulation parameters are set as follows, unless otherwise is stated. We assume that the signal of interest is of length $N = 200$ with sparsity level of $\lambda = \frac{K}{N} = 0.1$. The variance of noise, i.e., σ_n^2 , is set to have a signal-to-noise ratio (SNR) of 20 dB. Other model parameters are set as $\rho = 0.2$, $p_{01} = 0.02$, $\sigma_L = 10$, and $T = 30$.

To detect the ROI, i.e., support of the signal, after determining the estimate of the signal $\hat{\mathbf{x}}^{(t)}$, a simple thresholding is performed. Specifically, $\alpha_n^{(t)}$ is set to 1, if $\hat{x}_n^{(t)} \geq 1$.

Figure 7.3 shows the evolution of the inferred importance levels, i.e., \bar{c}_n , over time for $n = 1, 2, \dots, 30$. In other words, this figure illustrates how the sensing energy is distributed among the coefficients at each time step. As it is clear, at the first time step, $\bar{c}_n = 0.5, \forall n$, indicating unbiased estimate of importance levels when no further information is available. However, as more measurements are received, uncertainty decreases and the support of the signal is revealed. It is also worthwhile to point out that an error in the ROI detection procedure can potentially impact up

to $W = 5$ time slots. Error propagation, as well as computational complexity, are the main reasons that choosing large values for W should be avoided.

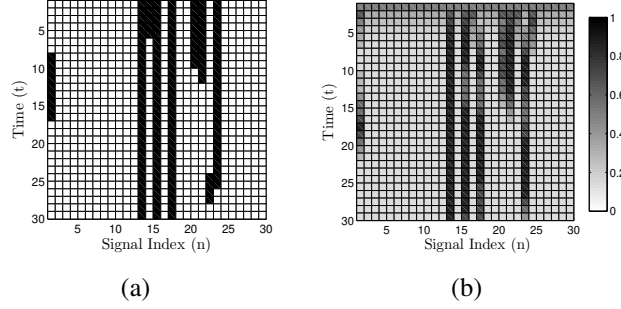


Figure 7.3: (a) Support of the signal and (b) the expected value of the inferred importance levels, i.e., \bar{c}_n , for the first 30 coefficients of the signal. $M = 60$, $N = 200$, $\text{SNR} = 20$ dB, and $W = 5$.

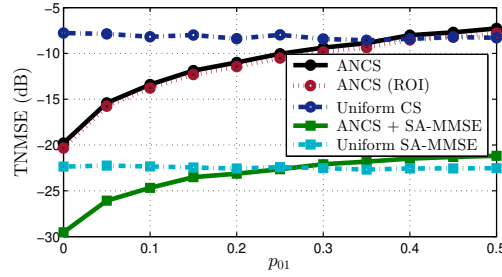


Figure 7.4: Performance of ANCS for different values of p_{01} . The total sensing energy is the same for all the methods. $N = 200$, $\text{SNR} = 20$ dB, and $T = 30$, and $M = 60$.

To study the performance of ANCS for different levels of temporal correlation, Figure 7.4 illustrates the TNMSE of ANCS for different values of p_{01} . The results are averaged over substantially large number of Monte-Carlo Trials. Here, ANCS is employed also as the sampling step of Support-aware MMSE, as well as the ℓ_1 minimization recovery method. SA-MMSE calculates the minimum mean square error estimate of the signal when the support of the sparse signal is known. The actual support of the signal, σ_L^2 , σ_n^2 , and ρ are provided as the inputs of the SA-MMSE algorithm. The performance of SA-MMSE is an indicator of lowest MSE achievable by a recovery algorithm.

For small values of p_{01} , the signal is nearly static over time. Thus, the method is able to detect the support accurately and the TNMSE is decreased significantly. Furthermore, since the signal is sparse in canonical basis and the support of the signal is set to be the ROI, overall recovery error is the same as the recovery error of the ROI coefficients. It is due to the fact that whole energy of the signal is concentrated in the ROI. As it can be noticed in the figure, for $p_{01} = 0$, ANCS can enhance the performance of the ℓ_1 minimization algorithm substantially. As p_{01} increases, the support of the signal changes over time and the observations of previous time steps become less informative about the signal and the performance gain of ANCS decreases. However, for values of $p_{01} < 0.3$, nonuniform recovery of the signal is still achieved.

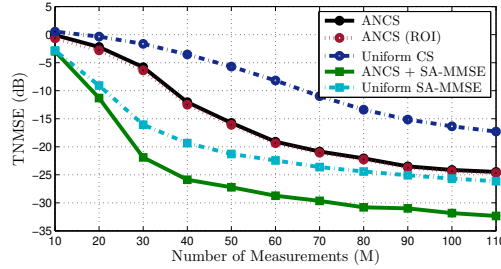


Figure 7.5: TNMSE (in dB) for of different recovery algorithm with and without ANCS as the sampling step for $N = 200$, $\text{SNR} = 20$ dB, and $T = 30$.

Figure 7.5 compares the performance of different recovery algorithms with uniform sampling and ANCS as the sampling step for different number of measurements. As it is clear, ANCS can decrease the TNMSE up to 7 dB, compared to ℓ_1 minimization recovery, and can reduce the required number of measurements. As an example, to achieve a TNMSE of -15 dB, ANCS employs about 50% of the measurements required by uniform sampling, highlighting one of the major benefits of ANCS: for a sparse signal in canonical basis, ANCS is able to reduce the recovery error and number of required measurements substantially.

To highlight the performance gain achieved by ANCS in low SNR regimes, Figure 7.6 depicts

TNMSE of different methods versus SNR. It is easy to notice that the performance of ANCS is very close to SA-MMSE with uniform sampling, which is MSE-optimal. This is one of the main benefits of adaptive CS. It is known that adaptive CS provides the opportunity to detect and estimate signals at lower SNRs. Furthermore, performance of SA-MMSE algorithm in Figure 7.5 and Figure 7.6 illustrates that ANCS is able to reduce the lower bound of recovery error by up to 6 dB.

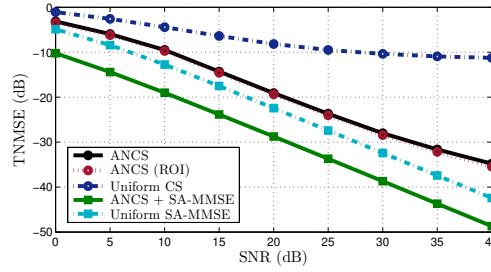


Figure 7.6: Performance of ℓ_1 minimization and SA-MMSE with and without ANCS as the sampling step in terms of TNMSE (in dB). of different methods for $N = 200$, $p_{01} = 0.02$, and $M = 60$.

7.4.2 Performance evaluation for sparse signals in the DCT domain

In this series of experiments, the performance gain achieved by ANCS is evaluated for signals that are not sparse in canonical basis, but has a sparse representation in some proper domain. In our numerical experiments, we employed DCT domain as the sparsifying basis.

To generate the sparse signal in DCT domain, the same procedure explained in Section 7.4.1 is exploited. Specifically, let $\mathbf{u}^{(t)} = \Psi \mathbf{x}^{(t)}$ represent the sparse representation of the signal of interest, $\mathbf{x}^{(t)}$, in DCT domain. Ψ denotes the DCT transform matrix. To generate a time correlated signal, elements of $\mathbf{u}^{(t)}$ are constructed as $u_n^{(t)} = s_n^{(t)} a_n^{(t)}$, where $s_n^{(t)}$ and $a_n^{(t)}$ are outcome of two random

processes described in Section 7.4.1. To reconstruct the signal, we use $\hat{\mathbf{x}}^{(t)} = \Psi^T \hat{\mathbf{u}}^{(t)}$, where

$$\hat{\mathbf{u}}^{(t)} = \arg \min \|\mathbf{u}\|_1, \text{ s.t. } \|\mathbf{y}^{(t)} - \Xi^{(t)}\mathbf{u}\|_2 < c,$$

and $\Xi^{(t)} = \Phi^{(t)}\Psi^T$.

Furthermore, to model the variation of ROI over time, a new set of binary Markov processes is employed. This means that the probability of a coefficient being in the ROI is independent from its location and its value. To describe this Markov process, for simplicity, we use the same set of parameters as the random process corresponding to the support of the signal, i.e., λ and p_{01} . Hence, the rates of change for support of $\mathbf{u}^{(t)}$ and the ROI in $\mathbf{x}^{(t)}$ are assumed to be the same. It is also assumed that the ROI detection algorithm may report erroneous observations to ANCS.

In Figure 7.7, we evaluate the performance of ANCS versus the number of measurements M for fault rate of 10%. This experiment also shows that the proposed ANCS is able to decrease the error of ROI coefficients up to 3-4 dB for different number of measurements. This benefit comes at the cost of losing performance on total recovery error. Interestingly, for smaller values of M , this benefit comes at almost no cost and without losing any performance for non-ROI entries.

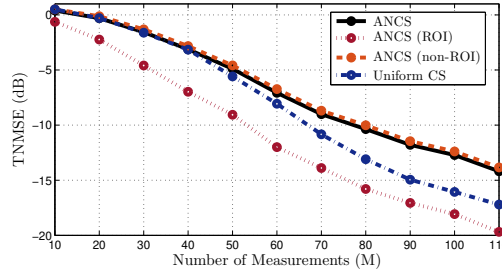


Figure 7.7: TNMSE (in dB) versus M of ANCS. $N = 200$, $\text{SNR} = 20$ dB, and $T = 30$, and $M = 60$.

Finally, as it was expected, Figure 7.8 illustrates that as ANCS receives more faulty data from the

ROI detection algorithm, its performance becomes more similar to conventional CS with uniform sampling. This is because the faulty data prevents the inference algorithm from gaining certainty on the location of ROI coefficients. However, even for fault rates of as much as 50%, non-uniform recovery of the signal is achieved.

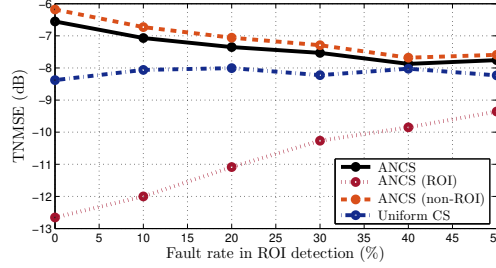


Figure 7.8: Plot of TNMSE (in dB) of ANCS for different values of fault rate. $N = 200$, $\text{SNR} = 20$ dB, and $T = 30$, and $M = 60$.

7.5 conclusions

This chapter presented *adaptive non-uniform compressive sampling* (ANCS) for time-varying sparse signals. The main idea is to employ the observations of previous time slots to infer the region of interest (ROI) in the signal and concentrate the sensing energy on the corresponding coefficients. For that, we presented a Bayesian framework, by modeling the overall and coefficient-specific reliability of the ROI detection algorithm.

The results show that the proposed framework is able to achieve the desired non-uniform recovery and can decrease the error in ROI significantly for signals that are sparse or have a sparse representation in a proper basis. The results also illustrated that the proposed method is particularly advantageous for signals that are sparse in canonical basis. For such signals, ANCS results in substantial improvement in accuracy of estimation.

CHAPTER 8: MISSING SPECTRUM-DATA RECOVERY IN COGNITIVE RADIO NETWORKS USING PIECEWISE CONSTANT NONNEGATIVE MATRIX FACTORIZATION

Recent advances in wireless communications and microelectronic devices are leading the trend of research toward cognitive radios (CRs) [153]. The main feature of CRs is the opportunistic usage of spectrum. CR systems try to improve the spectrum efficiency by using the spectrum holes in frequency, time, and space domains [153, 154]. This means that secondary users (SUs) are allowed to utilize the spectrum, provided that their transmissions do not interfere with the communication of primary users (PUs) [155]. The fundamental components of CR systems that allow them to avoid interference are spectrum sensing and resource allocation¹.

However, in a practical CR network, spectrum occupancy measurements for all the frequency channels at all times are not available. This is partially because of energy limitations and network failures. Another highly important and very common reason for occurrence of missing entries in the data set is the hardware limitation. Each SU may want to use different frequency channels, but it may not be capable of sensing all the channels simultaneously [157, 158]. On the other hand, a complete and reliable spectrum sensing data set is needed for a reliable resource allocation. Therefore, we need to develop a method to estimate the missing spectrum sensing measurements. This task is especially more challenging in dynamic environments.

There are different approaches toward the problem of data analysis in the CR networks. In [159], a learning approach is introduced based on support vector machine (SVM) for spectrum sensing in

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, M. Joneidi, B. Shahrabi, and N. Rahnavard, "Missing spectrum-data recovery in cognitive radio networks using piecewise constant Nonnegative Matrix Factorization," in *Proceedings - IEEE Military Communications Conference MILCOM*, vol. 2015-Decem, pp. 238–243, IEEE, 10 2015, © 2015 IEEE [156].

multi-antenna cognitive radios. SVM classification techniques are applied to detect the presence of PUs. Several algorithms have been proposed using dictionary learning framework [160, 161]. These approaches try to find the principal components of data using dictionary learning and exploit the components to extract information.

The goal of this chapter is to estimate the missing spectrum sensing data as accurate as possible in the time varying environments. An approach is introduced based on Nonnegative Matrix Factorization (NMF) [162, 163] to represent the spectrum measurements as additive, not subtractive, combination of several components. Each component reflects signature of one PU, therefore the data can be factorized as the product of signatures matrix times an activation matrix.

Dimension reduction is an inevitable pre-processing step for high dimensional data analysis. NMF is a dimension reduction technique that has been employed in diverse fields [164, 165]. The most important feature of NMF, which makes it distinguished from other component analysis methods, is the non-negativity constraint. Thus the original data can be represented as additive combination of its parts.

In our proposed method, a new framework is introduced to decompose the spectrum measurements in CR networks using a piecewise constant NMF algorithm in presence of missing data. Piecewise constant NMF and its application in video structuring is introduced in [166]. In the proposed method, we try to handle the missing entries in the data and also take a different approach to solve the corresponding optimization problem using an iterative reweighed technique.

In the context of CR networks, NMF is utilized in [167] to estimate the power spectra of the sources in a CR network by factorizing the Fourier Transform of the correlation matrix of the received signals. Our proposed method estimates the missing entries in power spectral density measurements by enforcing a temporal structure on the activity of the PUs and can be used in scenarios when the number of the PUs is not known.

The introduced method takes advantage of a prior information about the activity of the PUs and exploits piecewise constant constraint to improve the performance of the factorization. Moreover, a solution for the introduced minimization problem is suggested using the Majorization-Minimization (MM) framework.

The rest of the chapter is organized in the following order. In Section 8.1, the system model and the problem statement are introduced. Section 8.2 describes the proposed new NMF problem. In Section 8.3, a method is presented to solve the piecewise constant NMF problem in MM framework with missing data. Section 8.4 presents the simulation results and finally Section 8.5 draws conclusions.

8.1 System Model

Due to the nature of wireless environments, trustworthy information cannot be extracted from measurements of a single SU. To find the spectrum holes in frequency, time, and space, there exists a fusion center that collects and combines the measurements from all the SUs [157]. Cooperative spectrum sensing makes the missing data estimation algorithm more robust. Fusion center predicts the missing entries by using the collected measurements. However, since each SU is not able to sense the whole spectrum all the time, the data set collected from the SUs contains missing entries. Network failures, energy limitations, and shadowing can also cause loss of data.

Without loss of generality, we want to reconstruct the power map in a single frequency band. The network consists of N_P primary transmitters and N_R spectrum sensors that are randomly spread over the entire area of interest. Figure 8.1 illustrates an example of a network with $N_P = 2$ PUs and $N_R = 10$ SUs in a 100×100 area.

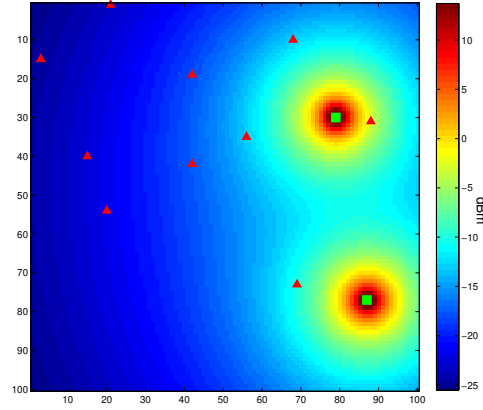


Figure 8.1: The power distribution of 2 PUs (squares) and deployment of 10 SUs (triangles) without considering shadowing effect.

The received power of the r^{th} sensor at time t can be written as

$$s_r(t) = \sum_{j=1}^{N_P} p_j(t) \gamma_{rj}(t) + z_r(t), \quad (8.1)$$

where $p_j(t)$ is the transmit-power of the j^{th} PU at time t , γ_{rj} is the channel gain from the j^{th} PU to the r^{th} SU, and $z_r(t)$ is the zero-mean Gaussian measurement noise at the r^{th} sensor with variance σ_r^2 . Considering a Rayleigh fading model, the channel gain coefficient can be modeled as:

$$\gamma_{rj} = \frac{C |h_{rj}|^2}{d_{rj}^\alpha}, \quad (8.2)$$

where the channel constant $C = \frac{G_P G_R c^2}{(4\pi f)^2}$, f is the carrier frequency, c is the speed of light, and G_P and G_R are the transmitter and receiver antenna gains. α is the path loss exponent which determines the rate at which power decays with the separation distance d_{rj} between the r^{th} SU and the j^{th} PU and $|h_{rj}|^2$ models the fading effect.

At time slot t , measurements from SUs can be stacked in a vector $\mathbf{s}(t)$, given as

$$\mathbf{s}(t) = \sum_{j=1}^{N_P} p_j(t) \boldsymbol{\gamma}_j(t) + \mathbf{z}(t), \quad (8.3)$$

where $\mathbf{s}(t) = \begin{bmatrix} s_1(t) & s_2(t) & \dots & s_{N_R}(t) \end{bmatrix}^T$, $\boldsymbol{\gamma}_j(t) = \begin{bmatrix} \gamma_{1j}(t) & \gamma_{2j}(t) & \dots & \gamma_{N_R j}(t) \end{bmatrix}^T$, and $\mathbf{z}(t) = \begin{bmatrix} z_1(t) & z_2(t) & \dots & z_{N_R}(t) \end{bmatrix}^T$. At each time slot, only a few SUs observe the power levels and report them to the fusion center. Therefore the vector $\mathbf{s}(t)$ contains some *missing* entries. Furthermore, each PU can be active or inactive in each time slot.

Some of the characteristics of the environment can be exploited to simplify the problem. It is assumed that channel gains are slowly time varying such that they can be considered as constant in a time window. Therefore, matrix representation of (8.3) can be written as:

$$\mathbf{S} = \mathbf{\Gamma} \mathbf{P} + \mathbf{Z}, \quad (8.4)$$

where \mathbf{S} is an $N_R \times T$ matrix, which includes measurements from sensors in T time slots, $\mathbf{\Gamma} = [\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_{N_P}]$ is a $N_R \times N_P$ matrix, which consists of N_P channel gain vectors in the N_R -dimensional space of data, and \mathbf{P} is an $N_P \times T$ matrix that indicates the power levels of PUs in each time slot ($p_{jt} = 0$ if the j^{th} PU is inactive at time t).

Here, the goal is to estimate the missing data using the partial observations. To achieve this goal, the data is decomposed using piecewise constant NMF. Then the components of data and the activation matrix are used to estimate the missing data.

8.2 PC-NMF: Piecewise Constant Nonnegative Matrix Factorization

Promoted by (8.4), it is easy to see that the measurements of each time slot can be represented as an additive, not subtractive, combination of few vectors. This algebraic representation has a geometric interpretation. Figure 8.2 helps us to visualize the structure of data in a 3-dimensional space of data. In this figure, 3 SUs are measuring power levels in an area with 3 PUs. It is easy to notice that measurement vectors lie within a pyramid in the positive orthant with $N_P = 3$ edges proportional to γ_j . This is due to fact that all the points in the pyramid can be written as an additive combination of the edges.

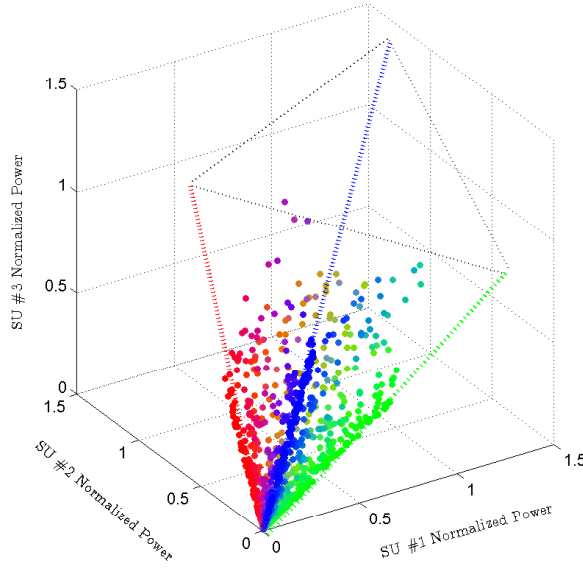


Figure 8.2: Structure of data generated by $N_P = 3$ PUs in $N_R = 3$ dimensional space.

Although it is assumed that the channel gains are stationary for a time window of length T , PUs can become activated/deactivated in this time window any number of times and can change their transmission power in each activation. Hence, the power levels of PUs tend to be piecewise con-

stant.

NMF is a widely-used technique to decompose data to its nonnegative components. Here, the structure of the power level matrix \mathbf{P} is exploited while handling the missing entries. As a result, the general objective function is presented as follows:

$$\begin{aligned} & \underset{\mathbf{\Gamma}, \mathbf{P}}{\text{minimize}} && D_W(\mathbf{S}|\mathbf{\Gamma}\mathbf{P}) + \beta F(\mathbf{P}), \\ & \text{subject to} && \mathbf{\Gamma} \geq 0, \mathbf{P} \geq 0, \end{aligned} \quad (8.5)$$

where $D_W(\mathbf{S}|\mathbf{\Gamma}\mathbf{P})$ is a weighted measure of fit and $F(\mathbf{P})$ is a penalty, which favors piecewise constant solutions. β is a nonnegative scalar weighting the penalty. The constraints denote that all the entries of $\mathbf{\Gamma}$ and \mathbf{P} are nonnegative. W is an $N_R \times T$ weight matrix that is used to estimate the weighted distance between \mathbf{S} and $\mathbf{\Gamma}\mathbf{P}$. The coefficients of the weight matrix denote the presence of data ($w_{rt} = 0/w_{rt} = 1$ if the measurement of the r^{th} SU at time slot t is unavailable/available).

NMF algorithms utilize different measures of fit such as Euclidean distance, generalized Kullback-Leibler (KL) divergence, and the Itakura-Saito divergence. In all the cases, the distance can be calculated as the sum of the distances between different coefficients [168–170].

$$D_W(\mathbf{S}|\mathbf{\Gamma}\mathbf{P}) = \sum_{t=1}^T \sum_{r=1}^{N_R} w_{rt} d(s_{rt} | \sum_{j=1}^{N_P} \gamma_{rj} p_{jt}), \quad (8.6)$$

In our case, Euclidean Distance is used as the measure of fit. This objective function is commonly used for problems with Gaussian noise model, a common noise model in communication systems, hence:

$$d(s_{rt} | \sum_{j=1}^{N_P} \gamma_{rj} p_{jt}) = \frac{1}{2} (s_{rt} - \sum_{j=1}^{N_P} \gamma_{rj} p_{jt})^2. \quad (8.7)$$

Since there exist sharp transitions in power level of PUs and power level of each PU is constant in

each transmission period, rows of P tend to be piecewise constant. In order to favor the piecewise constant solutions, the penalty function is defined as:

$$F(\mathbf{P}) = \sum_{t=2}^T \sum_{j=1}^{N_P} \lim_{n \rightarrow 0} |p_{jt} - p_{j(t-1)}|^n. \quad (8.8)$$

When n tends to 0, this penalty function represents the sum of ℓ_0 norm of the transition vectors, i.e. $\mathbf{p}_t - \mathbf{p}_{(t-1)}$, where \mathbf{p}_t is an $N_P \times 1$ vector containing power levels of PUs in time t . This penalty favors the solutions with a lower number of transitions. However, since it is not differentiable, it can be replaced with a differentiable approximation:

$$F_\epsilon(\mathbf{P}) = \sum_{t=2}^T \sum_{j=1}^{N_P} \rho_\epsilon(p_{jt} - p_{j(t-1)}), \quad (8.9)$$

with $\rho_\epsilon(x) = \frac{x^2}{x^2 + \epsilon^2},$

where ϵ^2 is a small positive constant and is much less than all the non-zero elements of $(p_{jt} - p_{j(t-1)})^2 \forall j, t$ to avoid division by zero.

In Section 8.3, an algorithm is derived to find the minimizer of the following problem:

$$\begin{aligned} & \underset{\Gamma, \mathbf{P}}{\text{minimize}} && D_W(\mathbf{S} | \Gamma \mathbf{P}) + \beta F_\epsilon(\mathbf{P}), \\ & \text{subject to} && \Gamma \geq 0, \mathbf{P} \geq 0. \end{aligned} \quad (8.10)$$

After estimating \mathbf{P} and Γ , the missing entries of \mathbf{S} can be approximated using the equation $\hat{\mathbf{S}} \simeq \Gamma \mathbf{P}$.

8.3 Majorization-Minimization for Piecewise Constant NMF

In this section, an iterative algorithm is described to find the solution of the optimization problem proposed in (8.10). For that, Majorization-Minimization (MM) framework is employed [168, 169]. MM algorithm and its variants have been used in various applications such as parameter learning and image processing [171, 172]. The update rules are derived to calculate the entries of \mathbf{P} given the entries of $\mathbf{\Gamma}$ and then the entries of $\mathbf{\Gamma}$ given the entries of \mathbf{P} , using an iterative reweighed algorithm.

First, the update rules for \mathbf{P} given $\mathbf{\Gamma}$ are derived. Then, the update rules for $\mathbf{\Gamma}$ will be derived in a similar manner.

As it is clear in (8.6), we can write the distance measure as a sum of different time slots:

$$D_W(\mathbf{S}|\mathbf{\Gamma}\mathbf{P}) = \sum_{t=1}^T C(\mathbf{p}_t), \quad (8.11)$$

where $C(\mathbf{p}_t)$ is the weighted Euclidean distance between \mathbf{s}_t and $\mathbf{\Gamma}\mathbf{p}_t$, given \mathbf{s}_t and $\mathbf{\Gamma}$. In MM framework, the update rules are derived by minimizing an auxiliary function [168]. By definition, $G(\mathbf{p}_t, \hat{\mathbf{p}}_t)$ is an auxiliary function of $C(\mathbf{p}_t)$ if and only if $G(\mathbf{p}_t, \hat{\mathbf{p}}_t) \geq C(\mathbf{p}_t)$ and $G(\mathbf{p}_t, \mathbf{p}_t) = C(\mathbf{p}_t)$ for $\forall \mathbf{p}_t$. If $G(\mathbf{p}_t, \hat{\mathbf{p}}_t)$ is chosen such that it is easier to minimize, the optimization of $C(\mathbf{p}_t)$ can be replaced with iterative minimization of $G(\mathbf{p}_t, \hat{\mathbf{p}}_t)$ over \mathbf{p}_t . Thus, in the literature, convex functions are frequently used as the auxiliary functions [164, 168]. It is shown in [168] that $C(\mathbf{p}_t)$ is non-increasing under the update

$$\mathbf{p}_t^{i+1} = \underset{\mathbf{p}_t}{\operatorname{argmin}} G(\mathbf{p}_t, \mathbf{p}_t^i). \quad (8.12)$$

This is due to the fact that in the i^{th} iteration we have $C(\mathbf{p}_t^{i+1}) \leq G(\mathbf{p}_t^{i+1}, \mathbf{p}_t^i) \leq G(\mathbf{p}_t^i, \mathbf{p}_t^i) = C(\mathbf{p}_t^i)$.

Following a similar approach as [168], the auxiliary function for the weighted Euclidean distance can be formulated as:

$$G(\mathbf{p}_t, \mathbf{p}_t^i) = C(\mathbf{p}_t^i) + (\mathbf{p}_t - \mathbf{p}_t^i) \nabla C(\mathbf{p}_t^i) + \frac{1}{2} (\mathbf{p}_t - \mathbf{p}_t^i)^T K(\mathbf{p}_t^i) (\mathbf{p}_t - \mathbf{p}_t^i), \quad (8.13)$$

where $K(\mathbf{p}_t^i)$ is an $N_P \times N_P$ diagonal matrix with

$$\begin{aligned} k_{jj}(\mathbf{p}_t^i) &= \frac{q_{jt}^i}{p_{jt}^i}, \\ \mathbf{q}_t^i &= \mathbf{\Gamma}^T(\mathbf{w}_t \odot \mathbf{\Gamma} \mathbf{p}_t^i), \end{aligned} \quad (8.14)$$

and k_{jj} is the j^{th} diagonal entry of $K(\mathbf{p}_t^i)$ and \odot is element-wise multiplication.

To solve the problem presented in (8.10), the contribution of $F_\epsilon(\mathbf{P})$ should be considered in the auxiliary function. For that, a convex version of $F_\epsilon(\mathbf{P})$ is employed:

$$\begin{aligned} F_\epsilon(\mathbf{P}) &= \sum_{t=2}^T \sum_{j=1}^{N_P} y_{jt} (p_{jt} - p_{j(t-1)})^2, \\ \text{with } y_{jt} &= \frac{1}{(p_{jt}^{(i-1)} - p_{j(t-1)}^{(i-1)})^2 + \epsilon}. \end{aligned} \quad (8.15)$$

Now the update rules can be obtained using the iterative version of (8.15). This means that y_{jt} is updated in each iteration using the values of \mathbf{P} in the previous iteration.

To form the penalized auxiliary function, $G_\beta(\mathbf{p}_t, \mathbf{p}_t^i)$, we add up $G(\mathbf{p}_t, \mathbf{p}_t^i)$ with the contribution of \mathbf{p}_t to $F(\mathbf{P})$. Thus, $G_\beta(\mathbf{p}_t, \mathbf{p}_t^i)$ can be written as:

$$G_\beta(\mathbf{p}_t, \mathbf{p}_t^i) = G(\mathbf{p}_t, \mathbf{p}_t^i) + \beta \left[\sum_{j=1}^{N_P} y_{jt}^i (p_{jt} - p_{j(t-1)}^i)^2 + \sum_{j=1}^{N_P} y_{j(t+1)}^i (p_{j(t+1)}^i - p_{jt})^2 \right]. \quad (8.16)$$

It is worthwhile to mention that $y_{j1} = y_{j(T+1)} = 0 \forall j$. Since $G_\beta(\mathbf{p}_t, \mathbf{p}_t^i)$ is convex, it can be easily minimized over \mathbf{p}_t by setting the gradient to zero. Hence the update rule is attained as:

$$p_{jt}^{i+1} = \frac{-\nabla_j C(\mathbf{p}_t^i) + p_{jt}^i k_{jj}(\mathbf{p}_t^i) + 2\beta y_{jt}^i p_{j(t-1)}^i + 2\beta y_{j(t+1)}^i p_{j(t+1)}^i}{k_{jj}(\mathbf{p}_t^i) + 2\beta y_{jt}^i + 2\beta y_{j(t+1)}^i}, \quad (8.17)$$

$$\nabla C(\mathbf{p}_t^i) = -\mathbf{\Gamma}^T(\mathbf{w}_t \odot \mathbf{s}_t - \mathbf{w}_t \odot (\mathbf{\Gamma} \mathbf{p}_t^i)),$$

where $\nabla_j C(\mathbf{p}_t^i)$ is the j^{th} element of the gradient $\nabla C(\mathbf{p}_t^i)$.

Finding the update rule for $\mathbf{\Gamma}$ is simple. This is due to the fact that $F(\mathbf{P})$ is not a function of $\mathbf{\Gamma}$. Hence, the update rule for $\mathbf{\Gamma}$ is similar to the update rule for standard NMF, except the missing entries must be taken into account [173]. The update rules can be written in matrix form as:

$$\mathbf{\Gamma}^{i+1} = \mathbf{\Gamma}^i \odot \frac{(\mathbf{W} \odot \mathbf{S}) \mathbf{P}^T}{(\mathbf{W} \odot (\mathbf{\Gamma}^i \mathbf{P})) \mathbf{P}^T} \quad (8.18)$$

where \odot is the element-wise multiplication and the division is also performed in an element-wise manner.

The obtained update rules in (8.17) and (8.18) are exploited alternatively to estimate $\mathbf{\Gamma}$ and \mathbf{P} . Then the missing entries of \mathbf{S} are predicted by $\hat{\mathbf{S}} = \mathbf{\Gamma} \mathbf{P}$.

However, by using the objective function in (8.10), the optimization problem results in solutions with entries of \mathbf{P} tend toward 0 and $\|\mathbf{\Gamma}\|$ tends toward ∞ . We take advantage of the scale ambiguity between $\mathbf{\Gamma}$ and \mathbf{P} to avoid this issue. Let $\mathbf{\Lambda}$ be a diagonal $N_P \times N_P$ matrix with its j^{th} diagonal entry equal to $\|\gamma_j\|_2$. In each iteration, the rescaled matrix pair $(\mathbf{\Gamma} \mathbf{\Lambda}^{-1}, \mathbf{\Lambda} \mathbf{P})$ is used instead of the original matrix pair $(\mathbf{\Gamma}, \mathbf{P})$.

As a practical scenario, we should also consider the case when the secondary network has no information about the number of the PUs, i.e. N_P . In this case, the common dimension of matrices

Γ and \mathbf{P} is not known. There have been some efforts in model order selection in NMF [170]. In the numerical experiments, $K > N_P$ is used as the common dimension to factorize the data in such conditions. This is only possible if the secondary network has some information about the upper bound of N_P .

8.4 Numerical Results

For the numerical experiments, one frequency channel is considered with $N_p = 3$ active PUs in the area. Figure 8.3 illustrates the topology of the network. Incomplete measurements are collected from $N_R = 20$ SUs.

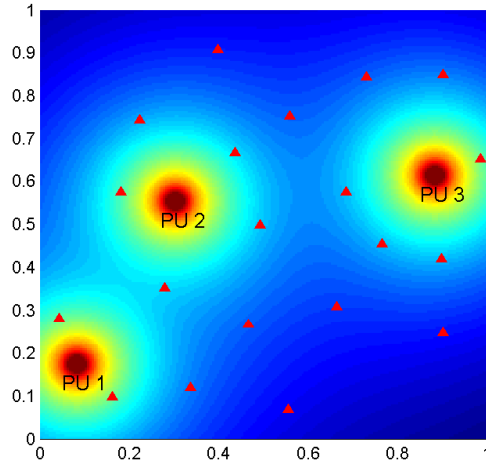


Figure 8.3: Network topology consisting of 3 PUs and $N_R = 20$ SUs marked by triangles.

We use the same simulation environment and the same network topology as in [161]. The simulation parameters are set as follows, unless otherwise is stated. The path loss is computed as $(\frac{d}{d_0})^\alpha$, where d is the distance, $d_0 = 0.01$, and $\alpha = 2.5$. γ_{rj} is computed by multiplying the pathloss by

the fading coefficient $|h_{rj}|^2$ where

$$h_{rj}(t) = \eta h_{rj}(t-1) + \sqrt{1-\eta^2} \nu_{rj}(t), \quad (8.19)$$

$\eta = 0.9995$, and $\nu_{rj}(t)$ is circularly symmetric zero mean complex Gaussian noise with variance 1 [161].

PUs' activity is modeled by a first order Markov model. All the PUs utilize the spectrum $\lambda = 0.3$ of the time slots. Transition matrix of the j^{th} PU is $\begin{pmatrix} 1-a_j & a_j \\ b_j & 1-b_j \end{pmatrix}$ and $\lambda = \frac{b_j}{a_j+b_j}$. a_j is the probability that the j^{th} PU stops transmitting from time $t-1$ to t and b_j is the probability that the PU activates transmitting. The parameter a_j is uniformly distributed over $[0.05, 0.15]$.

Each time a PU becomes activated, it chooses the transmission power from a uniform distribution with support $[100, 200]$. Each SU makes a measurement with 70% of chance. The measurements are contaminated by additive white Gaussian noise. The noise variance is 10^{-5} for all the SUs.

Partial measurements are generated for $T = 600$ time slots. To reduce the computational burden, the first 300 time slots are used to estimate Γ . Next, by using the obtained Γ and the update rule (8.17), \mathbf{P} is estimated for all 600 time slots. The regularization factor β is set to 5×10^{-3} and $K = 5$ factors are used to factorize the data.

Figure 8.4 shows the true power levels and the reconstructed one at a randomly selected SU versus time for the time window of $T = 300$ samples. It can be seen that the missing entries are accurately recovered through the proposed method, and it is evident that the proposed algorithm can easily track abrupt changes in power level.

Figure 8.5 compares the RMSE, averaged over SUs, of the proposed method with two similar

methods. The method introduced in [161] exploits the spatial correlation between adjacent SUs' measurements and semi-supervised dictionary learning (SS-DL) to estimate the missing entries. For the numerical results, the batch version of SS-DL is employed and the parameters are set to their optimal values. Furthermore, to emphasize the effect of the piecewise constant penalty, the results are also compared with the weighted NMF, i.e. WNMF [173, 174]. WNMF employs binary weight matrix to deal with the missing entries. This figure shows that the proposed method outperforms its competitors in different noise levels (Figure 8.5.(a)) and different probabilities of miss (Figure 8.5.(b)). P_{miss} denotes the ratio of the missing entries among the spectrum data.

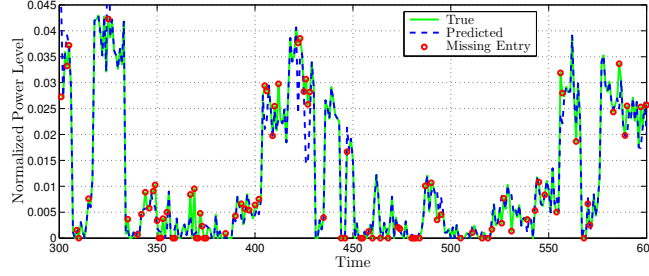


Figure 8.4: Power levels of a single SU.

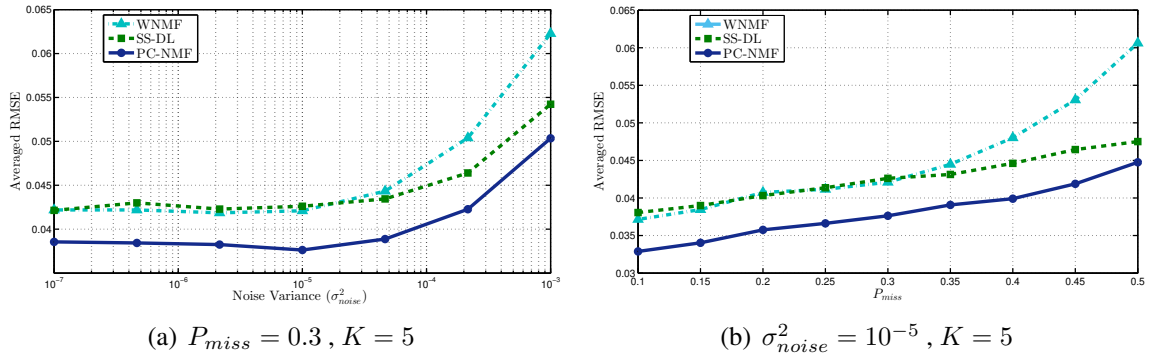


Figure 8.5: Performance of the proposed method for different noise levels and probability of miss, averaged over 200 Monte Carlo trials.

This figure shows that WNMF and SS-DL almost perform the same for low noise variance and low

P_{miss} . However, for harsh environments with high noise variance or high P_{miss} , SS-DL produces more accurate results. The PC-NMF method outperforms both methods in different noise levels and different probabilities of miss. For instance, PC-NMF has 11.6% less RMSE compared to the SS-DL method for $\sigma_{noise}^2 = 10^{-5}$ and $P_{miss} = 0.3$.

This improvement in the performance does not increase the computational burden of the algorithm. Table 8.1 shows the running times² for different methods averaged over 100 Monte Carlo trials for $\sigma_{noise}^2 = 10^{-5}$, $P_{miss} = 0.3$, and $K = 5$.

Table 8.1: Average Running Time

Method	Average Running Time (s)
SS-DL	10.3039
WNMF	0.0944
PC-NMF	0.0952

It is known that the NMF methods converge much faster than methods based on gradient descent [173]. However, Table 8.1 also illustrates that the proposed method does not require more computational resources compared with WNMF.

To study the effect of the piecewise constant penalty on the output of the algorithm, Figure 8.6 depicts the power level of two PUs and the estimated activation levels using the introduced method and WNMF. Both methods can estimate the power levels up to a scale factor. The number of factors is set to 3, i.e. $K = N_P$.

This figure illustrates the fact that the proposed method produces a more accurate factorization by taking advantage of piecewise constant constraint as a prior information. As it was expected, power

²All simulations have been performed under MATLAB 2014a environment on a PC equipped with Intel Xeon E5-1650 processor (3.20 GHz) and 8 GB of RAM.

levels estimated by PC-NMF are piecewise constant, while the results generated by WNMF are noisy. In fact, the piecewise constant penalty decreases the effect of noise and fading. Moreover, the sharp transitions are preserved in the factors returned by PC-NMF.

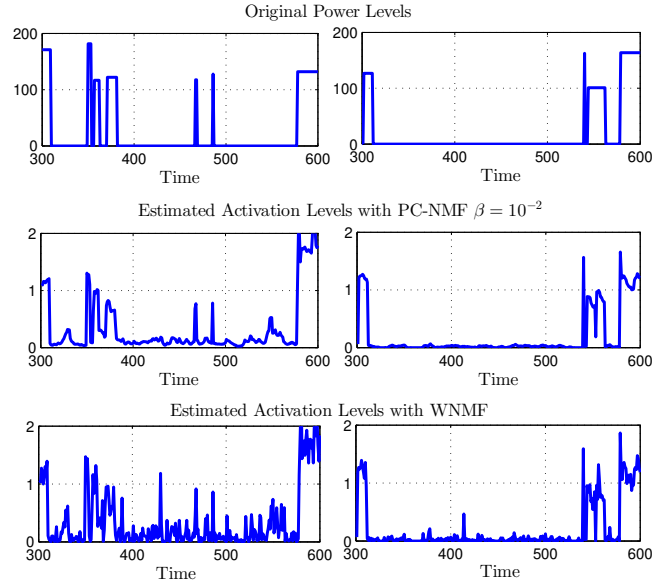


Figure 8.6: Original power levels of different PUs and the estimated activation levels with PC-NMF and WNMF.

8.5 Conclusions

By exploiting inherent structural feature of cognitive radio networks, we proposed a piecewise constant NMF approach that can decompose the data set into its components. Majorization-Minimization framework is utilized to solve the optimization problem of the piecewise constant NMF. Numerical simulations suggest that this method is able to predict the missing entries in the spectrum sensing database accurately.

CHAPTER 9: A BAYESIAN APPROACH FOR ASYNCHRONOUS PARALLEL SPARSE RECOVERY

Sparse recovery problems have received significant attention in the past decade, particularly in the compressed sensing (CS) literature [22, 23]. CS techniques have revolutionized sensing and sampling, with applications in image reconstruction [132, 175], hyper spectral imaging [176], wireless communications [177, 178], and analog to digital conversion [179]. Meanwhile, complex data-gathering devices have been developed, leading to the rapid growth of *big data*. For instance, the size of problems in hyperspectral imaging [176] are so large that they cannot be stored or solved in conventional computers. This, as well as the proliferation of inexpensive multi-processor computing systems, has motivated the study of parallel sparse recovery¹.

In parallel sparse recovery, the goal is to solve a large-scale sparse recovery problem by partitioning it among multiple processing nodes, thus reducing both the storage and computation requirements [181]. However, many recent studies [181–186] focus on *synchronous* parallel recovery of the sparse signal, meaning that some subset of the processing nodes need to wait for another subset of the nodes to complete their tasks. Of course, this approach is sensitive to slow or nonfunctional nodes.

Thus, it is natural to look for algorithms that divide the large-scale sparse recovery problems among several computing nodes and solve it *asynchronously*. Recently, in [187], the authors proposed a strategy to utilize the stochastic hard thresholding (StoIHT) [188, 189] in an asynchronous manner. Instead of sharing the current solution among the processors, which is the conventional approach

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, J. Haddock, N. Rahnavard, and D. Needell, “A Bayesian Approach for Asynchronous Parallel Sparse Recovery,” in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 2018-Octob, pp. 1980–1984, IEEE, 10 2019, © 2019 IEEE [180].

[181–183], an estimate of the support of the signal is shared. Then, the iteration number of each processor is used to assign weight to faster cores.

In this chapter, inspired by [187], we propose an asynchronous StoIHT [188] which incorporates a probabilistic framework that assigns reliability scores to each processor. This score is calculated by considering both the processor’s estimate of the support *and* its iteration number. Therefore, not only do we ignore the information from unreliable slow cores, but we are also able to utilize the reliable information from slower cores and disregard unreliable information from faster cores. The update rules for reliability scores and the support estimation is derived in a mathematical, less heuristic, manner, using variational inference [147]. This leads to simple closed form update rules for the parameters of the posterior distributions of the hidden variables with very low computational overhead. The convergence of our algorithm is theoretically analyzed in [190].

9.1 System Model

We consider the *sparse recovery* problem of reconstructing $\mathbf{x} \in \mathbb{R}^N$ from few nonadaptive, linear, and noisy measurements, $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{A} \in \mathbb{R}^{m \times N}$ is the measurement matrix and $\mathbf{z} \in \mathbb{R}^m$ is noise. One challenge to developing an asynchronous parallel approach to recovering the s -sparse signal \mathbf{x} via the optimization problem

$$\min_{\hat{\mathbf{x}} \in \mathbb{R}^N} \frac{1}{2m} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2^2 \quad \text{subject to} \quad \|\hat{\mathbf{x}}\|_0 \leq s$$

is that the cost function $\frac{1}{2m} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2^2$ is defined by the matrix \mathbf{A} , which is not generally sparse (e.g., standard i.i.d. Gaussian \mathbf{A} is common). A naive asynchronous approach would frequently overwrite the s non-zero entries learned by faster and more reliable processors. Our goal is to solve this problem in an asynchronous manner, while reducing the effects of *slow* processors on

the estimated signal. Note that the problem can be rewritten as

$$\min_{\hat{\mathbf{x}} \in \mathbb{R}^N} \frac{1}{M} \sum_{B=1}^M \frac{1}{2b} \|\mathbf{y}_B - \mathbf{A}_B \hat{\mathbf{x}}\|_2^2 \quad \text{subject to} \quad \|\hat{\mathbf{x}}\|_0 \leq s,$$

where \mathbf{y} and \mathbf{A} are partitioned into M non-overlapping sub-vectors \mathbf{y}_B and sub-matrices \mathbf{A}_B . At each iteration, each processor solves a subproblem by using the $b = m/M$ equations defined by \mathbf{A}_B and \mathbf{y}_B . We do not assume that the number of subproblems M and processors P is the same.

9.2 Probability Model

Our Bayesian algorithmic framework makes use of a *tally vector* $\boldsymbol{\phi} \in \mathbb{R}^N$ which records information on the current estimated support of \mathbf{x} . This vector and several reliability estimates are the *hidden variables* in our model:

1. *Tally score*, $\phi_n \in [0, 1]$, describing the probability that coefficient n is in support.
2. *Reliability score* for each processor, $r_i \in [0, 1]$, denoting the trustworthiness of the measurements of processor i .
3. *Observation reliability*, $u_{ni} \in \{0, 1\}$, which indicates if coefficient n reported by processor i is reliable.

Our estimates of these hidden variables are updated according to the following *observed variables*:

1. The *support observations*, o_{ni} which indicates if processor i detects coefficient n in the support.
2. The maximum *number of iterations* completed by any processor since the last reporting of processor i , k_i .

The posterior probability distribution of these hidden variables (referred to as \mathcal{H}) are inferred from the observed variables reported by the processors, o_{ni} and k_i (referred to as \mathcal{D}), according to the following generative model where variables are indexed for $i = 1, \dots, P$ and $n = 1, \dots, N$.

$$\begin{aligned}
r_i &\sim \text{Beta}(\beta_i^1, \beta_i^0) \\
u_{ni} &\sim \text{Bernoulli}(r_i) \\
\phi_n &\sim \text{Beta}(a_n^1, a_n^0) \\
o_{ni} &\sim u_{ni} \text{Bernoulli}(\phi_n) \\
&\quad + (1 - u_{ni}) \text{Bernoulli}(1 - \phi_n) \\
k_i &\sim \text{Binomial}(K_i, r_i)
\end{aligned} \tag{9.1}$$

The variable for the reliability score, r_i , is modeled with a Beta distribution with parameters β_i^1 and β_i^0 . This is the natural choice since r_i is used as the parameter of the Bernoulli distribution of the observation reliability and the conjugate prior for a Bernoulli distribution is Beta distribution.

The observation reliability is modeled as a Bernoulli distribution with parameter r_i . If a processor is generally reliable (r_i close to 1) it is more likely to be reliable on other coefficients.

The tally score is also defined as a random variable sampled from a Beta distribution with parameters a_n^1 and a_n^0 . This is also because ϕ_n is later used as the parameter of the Bernoulli distribution that describes the observations.

The observed variable, o_{ni} , is defined as the summation of two Bernoulli distributions. If an observation of the processor is reliable, $u_{ni} = 1$, the distribution would be $\text{Bernoulli}(\phi_n)$. This means that o_{ni} will be sampled from a Bernoulli distribution with true parameter, i.e., ϕ_n . By definition, ϕ_n is defined as the probability that coefficient n belongs to the support of the signal. Otherwise, for $u_{ni} = 0$, it will be sampled from $\text{Bernoulli}(1 - \phi_n)$, which means it reports faulty data.

Finally, k_i , the number of iterations completed by processor i is modeled with $\text{Binomial}(K_i, r_i)$ where K_i is the maximum number of iterations completed by any processor since the last reporting of processor i . Thus, we have $k_i \leq K_i$. Reliable processors (r_i close to 1) are likely to report k_i close to K_i .

The goal of our sequential Bayesian updating inference algorithm is to infer the distribution of \mathcal{H} given \mathcal{D} .

9.3 Inference via Sequential Bayesian Updating

Using Bayes' rule, the posterior distribution is

$$\mathbb{P}\{\mathcal{H}|\mathcal{D}\} \propto \mathbb{P}\{\mathcal{D}|\mathcal{H}\}\mathbb{P}\{\mathcal{H}\} = \mathbb{P}\{\mathcal{D}, \mathcal{H}\}.$$

where $\mathbb{P}\{\mathcal{D}, \mathcal{H}\}$ is calculated using the model described in (9.1). Specifically, the last two expressions in (9.1), are used to build $\mathbb{P}\{\mathcal{D}|\mathcal{H}\}$ and the other terms represent our prior belief, $\mathbb{P}\{\mathcal{H}\}$. The posterior distribution is the updated distribution of the hidden variables after receiving the observations.

In sequential Bayesian updating, the prior knowledge of the model is represented as the prior distribution, which is the distribution of the hidden variables before collecting data. After observing the first set of measurements, the posterior distribution is determined using Bayes' rule. Then, the posterior distribution can be used as the prior when the next set of observations becomes available. Thus, we must update the distribution of the hidden variables using the observations. In this approach, all the information is stored in the current distribution of the hidden variables.

To handle the intractable integrals arising in the inference procedure, *variational inference* is em-

ployed [146, 147]. In variational inference, the posterior distribution is approximated by a family of distributions for which the calculations are tractable. The approximate distribution is assumed to be fully factorized over all the hidden variables [149, Chapter 10]. Specifically, the fully factorized variational distribution $\mathbb{Q}\{\mathcal{H}\}$ is defined as

$$\mathbb{Q}\{\mathcal{H}\} = \prod_i \mathbb{Q}\{r_i | \hat{\beta}_i^1, \hat{\beta}_i^0\} \prod_{n,i} \mathbb{Q}\{u_{ni} | \tau_{ni}\} \prod_n \mathbb{Q}\{\phi_n | \hat{a}_n^1, \hat{a}_n^0\}, \quad (9.2)$$

where $\hat{\beta}_n^1$, $\hat{\beta}_n^0$, \hat{a}_n^1 , \hat{a}_n^0 , and τ_{ni} are the parameters of the factorized distributions. Our goal is to obtain $\mathbb{Q}\{\mathcal{H}\}$ such that it approximates the posterior distribution $\mathbb{P}\{\mathcal{H} | \mathcal{D}\}$.

Thus, at each step, the optimization problem

$$\max \quad \mathbb{E}\{\ln(\mathbb{P}\{\mathcal{D}, \mathcal{H}\})\} - \mathbb{E}\{\ln(\mathbb{Q}\{\mathcal{H}\})\}$$

(where the expected value is with respect to variational distribution) is solved with respect to one of the factorized distributions, keeping all other distributions fixed. The procedure is repeated until convergence. Each step results in a *closed form* update rule for one of the variables. Since the objective function is convex with respect to each of the factorized distributions, convergence is guaranteed [149, Chapter 10]. The derivations of the updating rules are presented in the Appendix E. After receiving each set of new measurements, the probability distributions of the unknown variables are updated using the closed form update rules. In this framework, the tally score for each coefficient, ϕ_n , is a random variable. Thus, the expected value of the random variable is used as a point estimate of the tally score and is denoted by

$$\bar{\phi}_n = \mathbb{E}_{\mathbb{Q}\{\phi_n\}}\{\phi_n\} = \frac{\hat{a}_n^1}{\hat{a}_n^1 + \hat{a}_n^0}. \quad (9.3)$$

Furthermore, we indicate the tally vector by $\phi = [\bar{\phi}_1, \bar{\phi}_2, \dots, \bar{\phi}_N]$. Details of the proposed frame-

work can be found in Algorithm 5 and the performance of the algorithm is evaluated in Section 9.4.

Algorithm 5 Bayesian Asynchronous StoIHT Iteration

Require: Number of subproblems, M , and probability of selection $p(B)$. The parameters of distribution of the reliability score, $\hat{\beta}_i^1$ and $\hat{\beta}_i^0$, and the parameters of tally scores, \hat{a}_n^1 and \hat{a}_n^0 , are available to each processor.

Each processor performs the following at each iteration:

- 1: **randomize:** select $B_t \in [M]$ with probability $p(B_t)$
 - 2: **proxy:** $\mathbf{b}^{(t)} = \mathbf{x}^{(t)} + \frac{\gamma}{Mp(B_t)} \mathbf{A}_{B_t}^* (\mathbf{y}_{B_t} - \mathbf{A}_{B_t} \mathbf{x}^{(t)})$
 - 3: **identify:** $\hat{\mathcal{S}}^{(t)} = \text{supp}_s(\mathbf{b}^{(t)})$ and $\tilde{T}^{(t)} = \text{supp}_s(\phi)$
 - 4: **estimate:** $\mathbf{x}^{(t+1)} = \mathbf{b}_{\hat{\mathcal{S}}^{(t)} \cup \tilde{T}^{(t)}}^{(t)}$
 - 5: **repeat**
 - 6: update $\mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\} = \mathbb{Q}\{u_{ni} = 1\}$ as described in Appendix E.3
 - 7: update $\hat{\beta}_i^1$ and $\hat{\beta}_i^0$ using (E.6), \hat{a}_n^1 and \hat{a}_n^0 using (E.5)
 - 8: **until** convergence
 - 9: update ϕ using (9.3)
 - 10: $t = t + 1$
-

It is clear that the inference algorithm is an iterative method. However, we will show in Section 9.4 that the framework performs well even if we run the update rules only once.

9.4 Numerical Experiments

9.4.1 Experiments in MATLAB

In these experiments, we take the signal dimension $N = 1000$, the sparsity level $s = 20$, and the number of measurements $m = 300$. Also, initial values for β_i^1 , β_i^0 , a_n^1 , and a_n^0 are set to 1, which results in uniform distribution for all r_i and ϕ_n and indicates unbiased estimate of processor reliability and tally score in absence of further information. For Stochastic IHT, the block size b is set to be same as the sparsity level s and $\gamma = 1$. The convergence criteria is $\|\mathbf{y} - \mathbf{A}\mathbf{x}^t\| \leq 10^{-7}$ and the maximum number of iterations is 1500.

Figure 9.1 shows the mean number of time steps over 50 trials when (a) all processors take the same amount of time to complete an iteration, and (b) half of the processors are *slow*, meaning that they complete an iteration every four time steps. It is evident that time steps required using the Bayesian update rules have decreased compared to standard non-parallel Stochastic IHT and Tally-based asynchronous IHT [187].

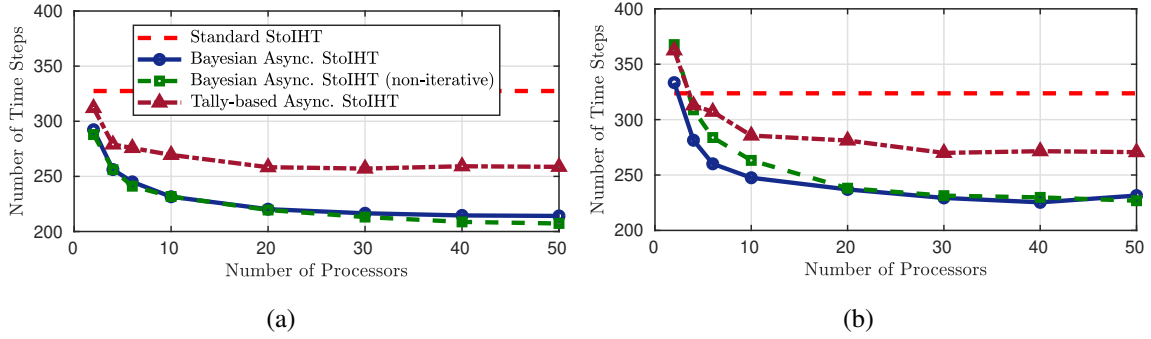


Figure 9.1: Comparison of the number of time steps until convergence versus the number of processors used in different methods, when (a) all processors complete an iteration in a single time step and (b) half of the processors complete an iteration every four time steps.

As mentioned in Section 9.3, the proposed inference algorithm is an iterative algorithm and needs to run the update rules alternatively to reach convergence. However, we also evaluate the performance of the non-iterative inference algorithm in which, at each StoIHT iteration, the inference runs the update rule for each variable only once. Figure 9.1 shows that the non-iterative and iterative algorithms performs similarly.

9.4.2 Experiments in C++

In this set of experiments, to have a better understanding of the behavior of the algorithms in a real parallel environment, different sparse recovery methods are implemented and tested using the C++ programming language and OpenMP [191], a multiprocessor shared memory programming

platform. Here, we take $N = 10000$, $m = 3000$, and $s = 200$. All other simulation parameters are same as Section 9.4.1. Running time reflects the time required to execute all the steps of the algorithms, including initialization, preprocessing, convergence, and post-processing. All simulations have been performed in the Ubuntu 16.04 environment on a PC equipped with an Intel Xeon E5-1650 processor (3.20 GHz) with 12 processors and 8 GB of RAM. Parallel AMP, a synchronous sparse recovery algorithm, is a row-wise multi-processor approximate message passing algorithm, as described in [181]. Here, we use the non-iterative version of the Bayesian asynchronous StoIHT algorithm. All the results are based on 50 Monte-Carlo trials.

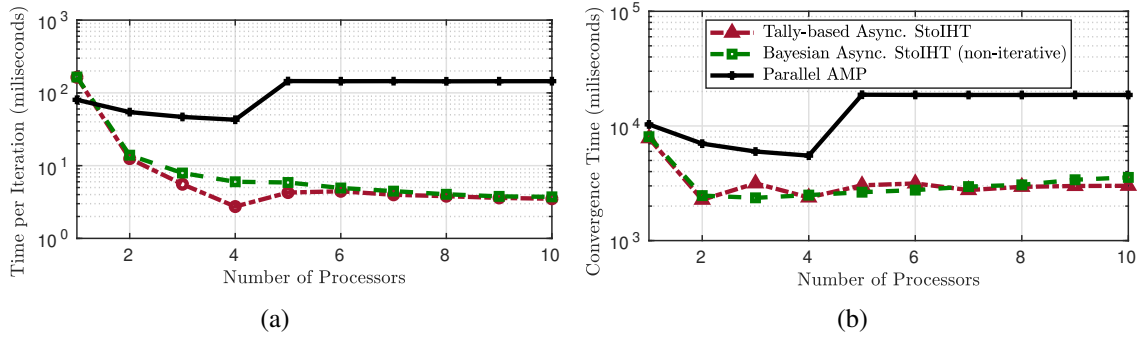


Figure 9.2: Performance of different multi-processor sparse recovery algorithm implemented using C++ programming language and OpenMP platform. Twenty percent of the processors are slow.

Figure 9.2(a) and Figure 9.2(b) show the execution time per iteration and total convergence time, respectively, for different numbers of processors. In this experiment, slow processors sleep for 100 ms at each StoIHT iteration and make up 20% of the processors; i.e., no slow processors for $P < 5$, one for $5 \leq P < 10$, and two for $P = 10$. After adding the first slow processor, the execution time for parallel AMP increases significantly, illustrating the fact that synchronous parallel algorithms suffer from the presence of slow processors. On the other hand, the execution time of the asynchronous algorithms does not change significantly. It is worthwhile to mention that at $P = 10$, when the second slow processor is added to the system, there is no increase in

the execution time of any of the algorithms, since more slow cores with the same sleep time does not increase the wait time of the system. Figure 9.2(b) shows that although the execution time per iteration of the asynchronous algorithms is decreasing after adding more processors, the total convergence time increases slightly, since more cores increases the processor/thread scheduling overhead and takes the inference algorithm longer to converge.

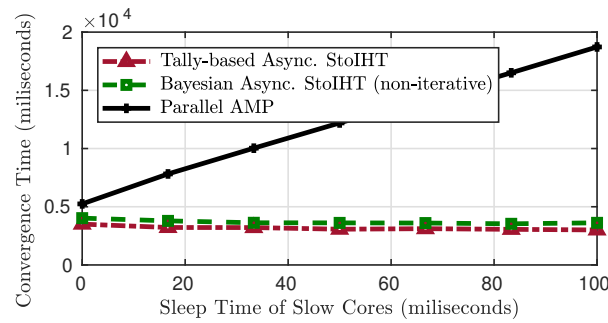


Figure 9.3: Mean convergence time of different multi-processor sparse recovery algorithms over 50 trials. Ten processors are solving the sparse recovery problem and two cores are slow.

Figure 9.3 demonstrates the effect of the sleep time of the two slow processors (of ten) on the execution time. Note that the convergence times of synchronous multi-processor algorithms increase linearly as the processors become slower, since all processors need to wait for slower processors at each iteration. The convergence time of the asynchronous algorithms depend less on the sleep time of the slow cores.

9.5 Conclusions

In this work, we modified the stochastic iterative hard thresholding algorithm to solve the sparse recovery problem in an asynchronous parallel manner. We proposed a Bayesian framework to assign reliability scores to the processing nodes, using both their current estimate of the support and their iteration number. The update rules for the reliability score and the support estimate are derived

in closed form using variational inference. This computationally inexpensive inference makes the algorithm more robust to slow, unreliable processing nodes. An interesting future direction is to utilize this framework for other sparse recovery algorithms such as AMP [192], which is known to have a better phase-transition threshold.

CHAPTER 10: ROBUST TARGET LOCALIZATION BASED ON SQUARED RANGE ITERATIVE REWEIGHTED LEAST SQUARES

The problem of localization arises in different fields of study such as wireless networks, navigation, surveillance, and acoustics [193–195]. There are many different approaches to localization based on various types of measurements such as range and squared-range (SR), time-of-arrival (ToA), time-difference-of-arrival (TDoA), two-way time-of-flight (TW-ToF), direction-of-arrival (DoA), and received-signal-strength (RSS) [194, 196–199]¹.

In [194], localization from range measurements and range-difference measurements are considered and least-squares (LS) estimators are exploited. Authors in [193–195, 199] have established methods to find the exact or approximate solution in the maximum likelihood (ML) framework. Usually finding the solution for ML estimators is a difficult task or computationally burdensome [195, 199].

In this chapter, the problem of *robust target localization* is considered. In sensor networks, some nodes may report faulty data to the processing node unintentionally or maliciously. This may occur because of network failures, low battery, physical obstruction of the scene, and attackers. Thus, the processing node should not simply aggregate measurements from all sensors. It is more efficient to disregard the outlier measurements and localize the target based on reliable measurements.

There are different approaches toward robust localization. The method in [196] is obtained by modeling the ToA estimation error as Cauchy-Lorentz distribution. In [200], robust statistics, and specifically Huber norm, is exploited to localize sensors in a network in a distributed manner using

¹Portions of this chapter is reprinted, with permission, from A. Zaeemzadeh, M. Joneidi, B. Shahrabi, and N. Rahnavard, “Robust Target Localization Based on Squared Range Iterative Reweighted Least Squares,” in *2017 IEEE 14th International Conference on Mobile Adhoc and Sensor Systems*, IEEE Computer Society, 2017, © 2017 IEEE [190].

the location of a subset of nodes. Authors in [198] try to minimize the worst-case likelihood function and employ semidefinite relaxation to attain the estimate using TW-ToF measurements. The authors in [201] have developed a robust geolocation method by estimating the probability density function (PDF) of the measurement error as a summation of Gaussian kernels. This method works best when the measurement error is drawn from a Gaussian mixture PDF.

In this chapter, the goal is to localize a single target in the presence of outlier range measurements in a centralized manner. We aim to achieve *outlier distributional robustness*, which means the estimator performs well for different outlier probability distributions. A least squares methodology is applied to the squared range measurements. Although, this formulation is not optimal in the ML sense [195], it provides us with the opportunity to find the estimate efficiently.

The contributions of this work can be summarized as follows. First, a robust optimization problem is formulated, which disregards unreliable measurements, using squared-range formulation. Next, two different algorithms are proposed to find the solution of the optimization problem. In the first algorithm, which is based on iteratively reweighted least squares (IRLS), the proposed optimization problem is transformed into a special class of optimization problems, namely Generalized Trust Region Subproblems (GTRS) [202]. Numerical simulations show that this algorithm has fast objective convergence. However the whole-sequence convergence is not established theoretically.

The second algorithm is based on gradient descent. This algorithm is globally convergent, but needs more iterations to converge. By using these two algorithms, we proposed a hybrid method, which has desirable theoretical and practical features, such as fast whole-sequence convergence.

The rest of this chapter is organized in the following order. In Section 10.1, the system model is introduced. Section 10.2 describes the robust localization problem and two methods to tackle the problem are presented. Section 10.3 presents the simulation results and finally Section 10.4 draws conclusions.

10.1 System Model

Since the problem of source localization arises in different fields such as wireless networks, surveillance, navigation, and acoustics, a general system model is exploited. In the generalized model, the system is comprised of R sensors, with known locations, and the location of the target is estimated using the range measurements reported by these sensors. A central processing node collects the measurements and computes the location of the target.

Each sensor reports a range estimate, denoted by r_i , given by

$$r_i = \|\mathbf{x} - \mathbf{a}_i\|_2 + v_i, \quad i = 1, \dots, R, \quad (10.1)$$

where $\|\cdot\|_2$ denotes Euclidean distance, $\mathbf{x} \in \mathbb{R}^n$ is the coordinates of the target, $\mathbf{a}_i \in \mathbb{R}^n$ is the location of the i^{th} sensor and v_i models the measurement error. It is clear that for the aforementioned applications $n = 2$ or 3 .

The measurement errors v_i are assumed to be independent and identically distributed random variables. To model the outlier measurements, a two-mode mixture PDF is assigned to the measurement errors, which can be written as:

$$p_V(v) = (1 - \beta)\mathcal{N}(v; 0, \sigma^2) + \beta\mathcal{H}(v). \quad (10.2)$$

In other words, measurement errors are drawn from the distribution $\mathcal{N}(v; 0, \sigma^2)$ with probability $1 - \beta$ or the distribution $\mathcal{H}(v)$ with probability β . $\mathcal{N}(v; 0, \sigma^2)$ models the measurement noise for the outlier-free measurements, which is assumed to be a zero mean Gaussian distribution with variance σ^2 , and $\mathcal{H}(v)$ models the outlier errors. Thus, the probability β denotes the ratio of outlier measurements to all the measurements, also known as the *contamination ratio*. The outlier

error distribution, $\mathcal{H}(v)$, is commonly modeled with a Uniform distribution [203, 204], a shifted Gaussian distribution [201, 205, 206], a Rayleigh distribution [206], or an exponential distribution [207]. However, it is worthwhile to mention that our proposed method does not rely on the distribution of $\mathcal{H}(v)$.

Here, the goal is to estimate \mathbf{x} using the measurements r_i $i = 1, \dots, R$, while disregarding the measurements from outlier sensors. The processing node has no information about the number of the outlier sensors and the distribution of outlier measurements. Moreover, it is assumed that all the reported measurements including the noisy and irrelevant measurements are positive. For that, we exploit robust statistics and propose methods to obtain the solution.

10.2 Robust Localization From Squared Range Measurements

In this section, a localization method is developed by applying robust statistics to the squared range measurements. Although this formulation is not optimal in the ML sense, unlike the methods based on range measurements, the solution can be attained easily.

The conventional square-range-based least squares (SR-LS) formulation is as follows [194]:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \sum_{i=1}^R (\|\mathbf{x} - \mathbf{a}_i\|_2^2 - r_i^2)^2. \quad (10.3)$$

It is clear that the problem stated in (10.3) is not convex. However, we can transform (10.3) into a special class of optimization problems by reformulating it as a constrained minimization problem given by [194, 202]

$$\begin{aligned} &\underset{\mathbf{x}, \alpha}{\text{minimize}} \quad \sum_{i=1}^R (\alpha - 2\mathbf{a}_i^T \mathbf{x} + \|\mathbf{a}_i\|^2 - r_i^2)^2, \\ &\text{subject to} \quad \|\mathbf{x}\|^2 = \alpha. \end{aligned} \quad (10.4)$$

It is worthwhile to mention that α is also an outcome of the optimization procedure, not a parameter to be set. In this formulation, the unreliable measurements from outlier sensors affect the accuracy of localization significantly. We plan to use robust statistics to decrease the sensitivity of the estimator to the common assumptions. Here, *robustness* signifies insensitivity to small deviation from the common assumption, which is the Gaussian distribution for noise. In (10.2), the parameter β represents the deviation from this assumption. Our goal is to deal with the unknown distribution $\mathcal{H}(v)$ and to achieve *distributional robustness*.

As described in [208], a proposed statistical procedure should have the following features. It must be *efficient*, in the sense that it must have an optimal or near optimal performance at the assumed model, i.e., the Gaussian distribution for noise. It must be *stable*, i.e., robust to small deviations from the assumed model. Also, in the case of *breakdown*, or large deviation from the model, a catastrophe should not occur. In the numerical experiments, we will look for these features in the proposed methods.

The general recipe to robustize any statistical procedure is to decompose the observations to fitted values and residuals [208]. In our proposed methods, we will try to find the residuals and re-fit iteratively until convergence is obtained. Each term of summation in (10.4) corresponds to the residual from a single sensor. These residuals can be exploited to re-fit the observations iteratively.

Specifically, we use the residuals to assign weights to each observation. If an observation is fitted to the model, it should have a larger weight in the procedure of decision making. Inspired by [209], we define the objective function as:

$$\mathcal{J}(\mathbf{y}, \mathbf{w}) = \sum_{i=1}^R w_i (\tilde{\mathbf{a}}_i^T \mathbf{y} - b_i)^2 + \sum_{i=1}^R \epsilon^2 w_i - \ln w_i, \quad (10.5)$$

where $\tilde{\mathbf{a}}_i^T = \begin{bmatrix} -2\mathbf{a}_i^T & 1 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} \mathbf{x} & \alpha \end{bmatrix}^T$, $b_i = r_i^2 - \|\mathbf{a}_i\|^2$, and $\mathbf{w} \in \mathbb{R}^R$ is the weight vector with $w_i > 0, \forall i$. The value of the parameter ϵ is a function of the standard deviation of the noise, we set $\epsilon = 1.34\sqrt{3}\sigma$ based on the discussion presented in Appendix F.

The first summation of the objective function (10.5) is the weighted version of the objective in (10.4). The other terms are added in such a way that result in the commonly used class of M-estimators known as Geman-McClure (GM) function [210, 211]. The aim of GM function is reduce the effect of large errors, by interpolating between ℓ_2 and ℓ_0 norm minimizations [210]. There are other M-estimators with similar behavior as the Geman-McClure such as Tukey, Welsch, and Cauchy estimators. These types of M-estimators are known to be more robust to large errors than Huber M-estimator [210]. The desirable feature of Huber function is the convexity, unlike all the other mentioned estimators. However, our numerical results show that the proposed algorithms perform well for different scenarios and different values of contamination ratio.

Our goal is to minimize $\mathcal{J}(\mathbf{y}, \mathbf{w})$ over \mathbf{y} and \mathbf{w} . Specifically, we are solving the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{y}, \mathbf{w}}{\text{minimize}} && \mathcal{J}(\mathbf{y}, \mathbf{w}), \\ & \text{subject to} && \mathbf{y}^T \mathbf{D} \mathbf{y} + 2\mathbf{f}^T \mathbf{y} = 0, \\ & && w_i > 0, \forall i, \end{aligned} \tag{10.6}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & 0 \end{bmatrix}, \mathbf{f} = \begin{bmatrix} \mathbf{0}_{n \times 1} \\ -0.5 \end{bmatrix}. \tag{10.7}$$

Our algorithms will exploit an alternative approach to update the weights and \mathbf{y} . We initialize by taking $w_i^{(0)} = 1, \forall i$. Then at the k^{th} iteration, the following optimization problem is solved to

update the value of \mathbf{y} :

$$\begin{aligned} \mathbf{y}^{(k+1)} = & \arg \min && \mathcal{J}(\mathbf{y}, \mathbf{w}^{(k)}), \\ \text{subject to} &&& \mathbf{y}^T \mathbf{D} \mathbf{y} + 2 \mathbf{f}^T \mathbf{y} = 0. \end{aligned} \quad (10.8)$$

Likewise, the weights are updated as follows:

$$\begin{aligned} \mathbf{w}^{(k+1)} = & \arg \min && \mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w}), \\ \text{subject to} &&& w_i > 0, \forall i. \end{aligned} \quad (10.9)$$

This problem is convex and the global minimizer can be obtained easily. As a result, the weights are given by:

$$\begin{aligned} w_i^{(k)} &= \frac{1}{(e_i^{(k)})^2 + \epsilon^2}, \\ \text{where } e_i^{(k)} &= \tilde{\mathbf{a}}_i^T \mathbf{y}^{(k)} - b_i. \end{aligned} \quad (10.10)$$

Choosing such weights is common in iteratively reweighted least square (IRLS) methods [156, 208, 210, 212, 213].

In robust statistics terms, the measurements are decomposed into the fitted values $\mathbf{y}^{(k)}$ and residuals $\mathbf{e}^{(k)}$ at each iteration k . Then, the residuals are exploited to tune the weights of the observations. For large residuals, i.e., $e_i \gg \epsilon$, each term of the first summation in (10.5), tends to 1. Similarly, for small residuals each term in summation tends to zero. In other words, we are minimizing the number of the observations with large residuals.

Now, two different approaches to find the solution of (10.8) are introduced. In the first approach, we show that (10.8) can be mapped into a special class of optimization problems known as Generalized Trust Region Subproblems (GTRS) [202]. Then at each iteration, the exact solution is derived by

employing the GTRS formulations. In the second approach, a method based on gradient descent is introduced to solve the problem. This method is not as computationally efficient as the first approach, but offers an array of desirable theoretical features.

10.2.1 The Squared Range Iterative Reweighted Least Squares (SR-IRLS) Approach

The optimization problem in (10.8) can be formulated in the matrix form as:

$$\begin{aligned} & \underset{\mathbf{y}}{\text{minimize}} \quad (\mathbf{A}\mathbf{y} - \mathbf{b})^T \mathbf{W}^{(k-1)} (\mathbf{A}\mathbf{y} - \mathbf{b}), \\ & \text{subject to} \quad \mathbf{y}^T \mathbf{D}\mathbf{y} + 2\mathbf{f}^T \mathbf{y} = 0, \end{aligned} \tag{10.11}$$

with

$$\mathbf{A} = \begin{bmatrix} -2\mathbf{a}_1^T & 1 \\ \vdots & \vdots \\ -2\mathbf{a}_R^T & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} r_1^2 - \|\mathbf{a}_1\|^2 \\ \vdots \\ r_R^2 - \|\mathbf{a}_R\|^2 \end{bmatrix}, \tag{10.12}$$

and $\mathbf{W}^{(k)}$ is a diagonal weighting matrix in the k^{th} iteration and $w_i^{(k)}$ is the i^{th} diagonal entry of $\mathbf{W}^{(k)}$, $i = 1, \dots, R$.

Note that in (10.11), a quadratic objective function is being minimized subject to a quadratic equality constraint. This special class of optimization problems is called Generalized Trust Region Subproblem (GTRS) [202]. The equality constraint makes this optimization problem non-convex. However, it is shown that the global solution of GTRS problems can be obtained efficiently [194, 202].

Theorem 10.1. *Let $q : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}$ be quadratics and assume $\{\mathbf{x} \in \mathbb{R}^n : c(\mathbf{x}) = 0\}$ is not empty. If*

$$\mathbf{v} \neq 0, \mathbf{v}^T \mathbf{C}\mathbf{v} = 0 \Rightarrow \mathbf{v}^T \mathbf{Q}\mathbf{v} > 0, \tag{10.13}$$

where

$$\mathbf{Q} = \nabla^2 q, \mathbf{C} = \nabla^2 c,$$

then the optimization problem $\min\{q(\mathbf{x}) : c(\mathbf{x}) = 0\}$ has a global minimizer.

Theorem 10.2. *Let $q : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}$ be quadratics and assume that $\min\{c(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} < 0 < \max\{c(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$ and $\nabla^2 c \neq \mathbf{0}$. A vector \mathbf{x}^* is a global minimizer of problem $\min\{q(\mathbf{x}) : c(\mathbf{x}) = 0\}$ if and only if $c(\mathbf{x}^*) = 0$ and there is a multiplier $\lambda^* \in \mathbb{R}$ such that the Kuhn-Tucker condition*

$$\nabla q(\mathbf{x}^*) + \lambda^* \nabla c(\mathbf{x}^*) = \mathbf{0}$$

is satisfied with

$$\nabla^2 q(\mathbf{x}^*) + \lambda^* \nabla^2 c(\mathbf{x}^*)$$

positive semidefinite.

Specifically, using Theorem 10.1 and the definitions of \mathbf{A} , $\mathbf{W}^{(k)}$, and \mathbf{D} , we can easily verify that (10.13) holds for the proposed optimization problem in (10.11). Thus, the optimization problem (10.11) has a global minimizer for all the iterations. Also by using Theorem 10.2, $\mathbf{y}^{(k)}$ is an optimal solution of (10.11) if and only if there exists $\lambda \in \mathbb{R}$ such that:

$$\begin{aligned} (\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda \mathbf{D}) \mathbf{y}^{(k)} &= \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{b} - \lambda \mathbf{f}, \\ \mathbf{y}^{(k)T} \mathbf{D} \mathbf{y}^{(k)} + 2 \mathbf{f}^T \mathbf{y}^{(k)} &= 0, \\ \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda \mathbf{D} &\succeq 0. \end{aligned} \tag{10.14}$$

The last expression means that $\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda \mathbf{D}$ is positive semidefinite. The first two equalities in (10.14) can be exploited to obtain a solution for λ , i.e. λ^* . To ensure that $\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda^* \mathbf{D}$

is positive semidefinite, it is easy to show that we need to seek for λ^* in the interval

$$\lambda^* \geq -\frac{1}{\lambda_1(\mathbf{D}, \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})}, \quad (10.15)$$

where $\lambda_1(\mathbf{D}, \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})$ is the largest generalized eigenvalue of the matrix pair $(\mathbf{D}, \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})$.

It is shown that if (10.13) holds, then $\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda \mathbf{D} \succeq 0$ for some $\lambda \in \mathbb{R}$ [202, Theorem 2.2].

Moreover, the resulting characteristic function needed to be solved to find λ^* is strictly decreasing over this interval [202, Theorem 5.2]. Thus, at each iteration, λ^* can be obtained using a bisection algorithm. The interval for starting point of the bisection algorithm is specified as (λ_l, ∞) , where $\lambda_l = \max\{-(\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})_{ii}, i = 1, \dots, n\}$ [202].

Then, \mathbf{y} is updated using the estimated λ^* . Algorithm 6 illustrates the procedure to calculate the estimate of (10.11) using the equations in (10.14). The convergence of the algorithm is analyzed in Theorem 10.3.

Algorithm 6 Calculating the SR-IRLS estimate

Require: $\mathbf{a}_i, \mathbf{r}_i$ for $i = 1, \dots, R, \epsilon$, maximum number of iterations $maxIter$, and the convergence tolerance Δ .

- 1: **Compute** $\mathbf{A}, \mathbf{b}, \mathbf{D}$, and \mathbf{f} using (10.12) and (10.7).
 - 2: **Initialize** $w_i^{(0)} = 1, \forall i$, and $k = 1$.
 - 3: **repeat**
 - 4: $\lambda_l = \max\{-(\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})_{ii}, i = 1, \dots, n\}$.
 - 5: **Find** λ^* : solve $\mathbf{y}(\lambda)^T \mathbf{D} \mathbf{y}(\lambda) + 2 \mathbf{f}^T \mathbf{y}(\lambda) = 0$ using a bisection algorithm in interval (λ_l, ∞) , where $\mathbf{y}(\lambda) = (\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A} + \lambda \mathbf{D})^{-1} (\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{b} - \lambda \mathbf{f})$.
 - 6: **Update** \mathbf{y} : $\mathbf{y}^{(k)} = \mathbf{y}(\lambda^*)$.
 - 7: **Update** $\mathbf{w}^{(k)}$ using (10.10).
 - 8: **until** Convergence, i.e., if $|\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)}) - \mathcal{J}(\mathbf{y}^{(k-1)}, \mathbf{w}^{(k-1)})| < \Delta$ or the maximum number of iterations $maxIter$ is reached.
-

Theorem 10.3. *The sequence $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\}$ generated by by Algorithm 6 converges to a constant value and every limit point of the iterates $\{\mathbf{y}^{(k)}, \mathbf{w}^{(k)}\}$ is a stationary point of (10.6).*

Proof. See Appendix G. □

Inspection of the algorithm reveals that the matrix inversions are only needed for $(n+1) \times (n+1)$ matrices, where n is the space dimension and is equal to 2 or 3. Thus the main computational burden of the algorithm stems from the matrix multiplications. The per iteration complexity of the algorithm is $\mathcal{O}(R^2)$. Similarly, the growth rate for the legacy least square problem is also $\mathcal{O}(R^2)$. Thus, the main computational burden of the SR-IRLS algorithm arises from the number of the iterations.

Our numerical experiments show that the SR-IRLS method needs a few iterations to solve the problem. The convergence of the objective is also proven in Appendix G. However, due to the lack of convexity, the standard convergence analysis tools cannot be used to show the convergence of the whole-sequence of the iterates. The problem becomes more difficult when the objective function is not a linear or a quadratic function of the previous iterates. Thus, in Appendix G, the convergence of a *subsequence* of the iterates to a critical point is proved, although the whole-sequence convergence is almost always observed.

This motivates us to propose a globally convergent algorithm. In Section 10.2.2, an algorithm, referred to as SR-GD, is introduced to find the solution of (10.11) based on gradient descent. Then, we will integrate SR-IRLS and SR-GD to derive a *computationally efficient* and *globally convergent* algorithm.

10.2.2 The Squared Range Gradient Descent (SR-GD) Approach

In this section, a new algorithm for solving the optimization problem in (10.8) is proposed based on gradient descent (SR-GD), for which the convergence of the whole-sequence of the iterates has been proven theoretically [214]. For that, the Lipschitz continuity of the gradient of the objective function as well as the special form of the objective and the constraint are employed. The numerical experiments show that this algorithm needs more iterations to converge than the SR-IRLS. Our goal

will be to employ SR-GD and SR-IRLS to propose a hybrid fast converging algorithm.

Inspired by [214], at each iteration, the value of $\mathbf{y}^{(k)}$ is updated as follows:

$$\begin{aligned} \mathbf{y}^{(k)} = \arg \min_{\mathbf{y}} \quad & \langle \nabla_{\mathbf{y}} \mathcal{J}(\hat{\mathbf{y}}^{(k)}, \mathbf{w}^{(k-1)}), \mathbf{y} - \hat{\mathbf{y}}^{(k)} \rangle \\ & + l^{(k)} \|\mathbf{y} - \hat{\mathbf{y}}^{(k)}\|_2^2, \\ \text{subject to} \quad & \mathbf{y}^T \mathbf{D} \mathbf{y} + 2\mathbf{f}^T \mathbf{y} = 0, \end{aligned} \tag{10.16}$$

where

$$\hat{\mathbf{y}}^{(k)} = \mathbf{y}^{(k-1)} + \omega^{(k)}(\mathbf{y}^{(k-1)} - \mathbf{y}^{(k-2)}),$$

and $l^{(k)}$ is the Lipschitz constant of $\nabla_{\mathbf{y}} \mathcal{J}(\mathbf{y}, \mathbf{w}^{(k-1)})$ at the k^{th} iteration. By the definition of Lipschitz continuity, we have

$$\|\nabla_{\mathbf{y}} \mathcal{J}(\mathbf{u}, \mathbf{w}^{(k-1)}) - \nabla_{\mathbf{y}} \mathcal{J}(\mathbf{v}, \mathbf{w}^{(k-1)})\| \leq l^{(k)} \|\mathbf{u} - \mathbf{v}\|.$$

Intuitively, the first term of the objective finds the steepest descent, while the second term prevents large changes in the magnitude of the gradient. The Lipschitz constant of the gradient function limits the step size of the algorithm and the new estimate $\mathbf{y}^{(k)}$ is enforced to be around the prediction $\hat{\mathbf{y}}^{(k)}$. The prediction is constructed using the previous iterates and an extrapolation factor $\omega^{(k)} = \frac{1}{12} \sqrt{\frac{l^{(k-1)}}{l^{(k)}}}$ [214]. The update rule for \mathbf{w} is the same as (10.10).

This problem is not convex as well, but authors in [214] have proven the convergence of the whole-sequence of the algorithm by exploiting the properties of the objective.

It is easy to notice that the minimization problem stated in (10.16) is a GTRS problem. This is because a quadratic objective is minimized subject to a quadratic equality constraint. By exploiting the definition of \mathbf{D} and $l^{(k)}$, we can show that (10.13) holds. Thus the optimization problem in

(10.16) has global minimizer for all iterations. Also Theorem 10.2 states that $\mathbf{y}^{(k)}$ is an optimal solution of (10.16) if and only if there exists $\lambda \in \mathbb{R}$ such that:

$$\begin{aligned}
(l^{(k)} \mathbf{I}_{n+1} + \lambda \mathbf{D}) \mathbf{y}^{(k)} &= -\mathbf{A}^T \mathbf{W}^{(k-1)} (\mathbf{A} \hat{\mathbf{y}}^{(k)} - \mathbf{b}) \\
&\quad + l^{(k)} \hat{\mathbf{y}}^{(k)} - \lambda \mathbf{f}, \\
\mathbf{y}^{(k)T} \mathbf{D} \mathbf{y}^{(k)} + 2 \mathbf{f}^T \mathbf{y}^{(k)} &= 0, \\
\lambda &\geq \max \left\{ -l^{(k)}, -\frac{1}{\lambda_1(\mathbf{D}, \mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A})} \right\}
\end{aligned} \tag{10.17}$$

At each iteration, after finding the predicted value for the iterate $\hat{\mathbf{y}}^{(k)}$, the equality expressions in (10.17) are used to find λ and to update the values of \mathbf{y} and \mathbf{w} . We should look for the solution of λ in an interval that satisfies the positive semidefiniteness constraint. Since (10.13) holds, this interval exists and the characteristic function is strictly decreasing over this interval [202, Theorem 2.2, Theorem 5.2]. Algorithm 7 shows the steps to find the solution of the localization problem using the SR-GD method.

Algorithm 7 Calculating the SR-GD estimate

Require: $\mathbf{a}_i, \mathbf{r}_i$ for $i = 1, \dots, R$, ϵ , maximum number of iterations $maxIter$, and the convergence tolerance Δ .

- 1: **Compute** $\mathbf{A}, \mathbf{b}, \mathbf{D}$, and \mathbf{f} using (10.12) and (10.7).
 - 2: **Initialize** $\mathbf{W}^{(0)}$ with identity matrix, $\mathbf{y}^{(-1)} = \mathbf{y}^{(0)} = \mathbf{A}^\dagger \mathbf{b}$, $l^{(0)} = 0$, and $k = 1$.
 - 3: **repeat**
 - 4: $l^{(k)} = 2 \|\mathbf{A}^T \mathbf{W}^{(k-1)} \mathbf{A}\|_F$.
 - 5: $\omega^{(k)} = \frac{1}{12} \sqrt{\frac{l^{(k-1)}}{l^{(k)}}}$.
 - 6: $\hat{\mathbf{y}}^{(k)} = \mathbf{y}^{(k-1)} + \omega^{(k)} (\mathbf{y}^{(k-1)} - \mathbf{y}^{(k-2)})$.
 - 7: **Find** λ^* : solve $\mathbf{y}(\lambda)^T \mathbf{D} \mathbf{y}(\lambda) + 2 \mathbf{f}^T \mathbf{y}(\lambda) = 0$ using a bisection algorithm in interval $(-l^{(k)}, \infty)$, where $\mathbf{y}(\lambda) = (l^{(k)} \mathbf{I}_{n+1} + \lambda \mathbf{D})^{-1} (-\mathbf{A}^T \mathbf{W}^{(k-1)} (\mathbf{A} \hat{\mathbf{y}}^{(k)} - \mathbf{b}) + l^{(k)} \hat{\mathbf{y}}^{(k)} - \lambda \mathbf{f})$.
 - 8: **Update** \mathbf{y} : $\mathbf{y}^{(k)} = \mathbf{y}(\lambda^*)$.
 - 9: **Update** $\mathbf{w}^{(k)}$ using (10.10).
 - 10: **until** Convergence, i.e., if $\|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\| < \Delta$ or the maximum number of iterations $maxIter$ is reached.
-

The numerical experiments show that the SR-GD method needs more time to find the solution than SR-IRLS. This is due to the fact that in SR-GD, the value of the new iterate is bounded to be around the previous iterate, unlike the SR-IRLS method.

To take advantage of the fast convergence of the SR-IRLS and the whole sequence convergence of the SR-GD, we propose a hybrid method. Specifically, we can start with the SR-IRLS method and update the iterates by steps stated in Algorithm 6. After convergence of the objective function, which is proven in Appendix G, the update rules in Algorithm 7 are employed to find the final solution. The performance, convergence rate, and computational cost of this hybrid method is evaluated and compared with other methods in Section 10.3.

10.3 Numerical Results

In this section, we present the simulation results to evaluate the performance of our proposed methods. We will seek for the main features of a robust estimator, which are discussed in Section 10.3. We will examine the performance of the algorithms at the assumed model ($\beta = 0$), small deviations from model (small β), and large deviations from the model (large β). Moreover, we check distributional robustness of the proposed algorithms, which means that the performance of the methods will be evaluated for different outlier noise distributions $\mathcal{H}(v)$.

Two different simulation scenarios will be investigated. In Scenario I, a general system model is considered and the outlier measurements obey a uniform distribution, which models a harsh environment. In Scenario II, localization of a target in a cellular radio network is investigated. The geometry of sensors is taken from an operating network and the measurement errors are drawn from a Gaussian mixture distribution to model the non-line-of-sight (NLOS) measurements.

The performances of the proposed methods are compared with existing least-square-based [194]

and robust [201] methods.

10.3.1 Scenario I

The simulation parameters are as follows, unless otherwise is stated. In a $4000 \times 4000 \text{ m}^2$ area, there exist 10 sensors trying to localize a target. The sensors and the target are distributed uniformly at random, The range measurements are corrupted by the additive white Gaussian noise with standard deviation of $\sigma = 55 \text{ m}$. Moreover, among the sensors, there exist 4 outlier sensors. The noise of the outlier sensors are uniformly distributed in range $[-4000\sqrt{2}, 4000\sqrt{2}]$. Mathematically speaking, the distribution of the measurement error is as follows:

$$p_V(v) = (1 - \beta)\mathcal{N}(v; 0, \sigma^2) + \beta\mathcal{U}(v; -D_{max}, D_{max}), \quad (10.18)$$

where $\mathcal{U}(v; -D_{max}, D_{max})$ is a uniform distribution with support $[-D_{max}, D_{max}]$, which is modeling the outlier measurements. $\mathcal{N}(v; 0, \sigma^2)$ is a zero mean Gaussian distribution with variance σ^2 .

To ensure that all the range measurements are positive, we set the non-positive values equal to a small value, i.e. 10^{-5} . Localization is performed in a 2-dimensional space, i.e. $n = 2$.

The performances of the proposed methods SR-IRLS, SR-GD, and the hybrid version are compared with the performance of SR-LS [194], a least-square-based method, as well as a robust method, namely Robust Iterative Nonparametric (RIN) [201]. The performances are compared according to the root mean square error (RMSE),

$$\sqrt{\frac{1}{n}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}, \quad (10.19)$$

averaged over sufficiently large random simulations. \hat{x} is the estimated value of the target location x .

In our first numerical experiment, the convergence of SR-IRLS and SR-GD are compared. Figure 10.1 depicts $\frac{\|y^{(k)} - y^{(k-1)}\|}{\|y^{(k)}\|}$ at different iterations. Moreover, the labels show the elapsed time for some of the iterations. Although the convergence of the SR-GD method is theoretically provable, Figure 10.1 shows that it needs more iterations and more time to converge. The hybrid version of the algorithm (SR-Hybrid) uses the update rules of SR-IRLS until the convergence of the objective function, then it employs the update rules of SR-GD. As a result, it needs less iterations than SR-GD, while its convergence is still theoretically provable.

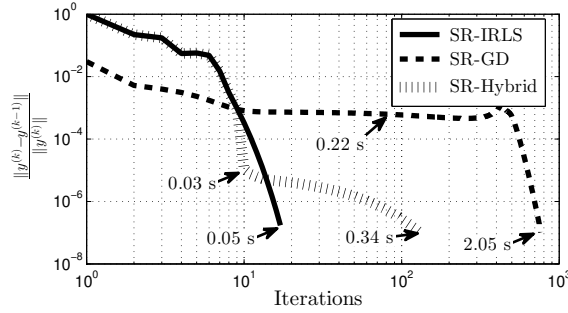


Figure 10.1: Convergence of SR-IRLS, SR-GD, and the hybrid method. Labels show the execution time of different algorithms at different iterations.

To study the influence of the number of outlier sensors, Figure 10.2 exhibits the RMSE of the estimate for different number of outlier sensors, or equivalently different values of β . In this study, the results are based on 200 Monte Carlo (MC) trials. It is clear that as the number of outliers increases, the performance of the SR-LS method deteriorates significantly. SR-IRLS and SR-GD perform closely for small values of β , but the difference becomes more noticeable as β increases. This was expected since SR-GD is more likely to result in local optimum solutions caused by the outliers, because of the smooth convergence of the iterates. However, the hybrid version, which only uses small step size when it is sufficiently close to the limit point, performs the best for

different values of contamination ratio. This figure shows that the proposed methods are efficient at the assumed method ($\beta = 0$) and stable for small deviations. Also, for large deviations, a catastrophe is not occurred.

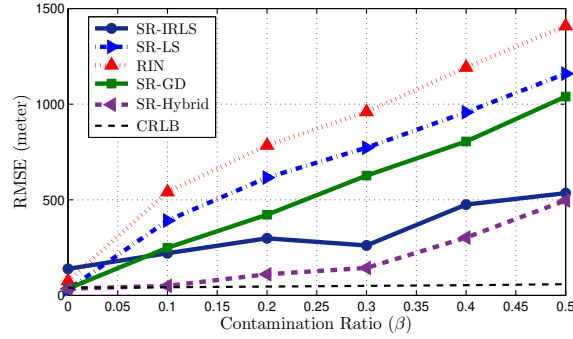


Figure 10.2: Robustness against outliers for 200 Monte Carlo trials, $\sigma = 55$ and $R = 10$. Number of outlier sensors is set to $\beta \times R$.

To estimate the target location, the RIN method [201] approximates the PDF of the measurement error with a summation of Gaussian kernels. For that, it needs a considerable number of measurements. Hence, unlike our proposed methods, it cannot produce proper results with $R = 10$ measurements. Further, since the RIN method employs Gaussian kernels, it works most accurately when the measurement errors are drawn from a Gaussian mixture distribution (see Section 10.3.2). Using the Gaussian kernels decreases the distributional robustness of the RIN method significantly.

To elaborate the point, Figure 10.3 illustrates the impact of the number of sensors on performance of different methods. In this experiment, 40% of the sensors are reporting unreliable data to the processing node, i.e. $\beta = 0.4$. This figure exhibits that the accuracy of the localization methods improves as the number of sensors increases. As it was expected, the RMSE of the estimates produced by the RIN method decreases significantly as the number of sensors increases.

Moreover, It is clear that the proposed methods meet the Cràmer-Rao lower bound (CRLB) for large number of measurements. From Figure 10.3(a) and Figure 10.3(b), we can infer that the pro-

posed methods are efficient for this simulation parameters, because they meet the CRLB and they are unbiased. The CRLB is approximated by using Monte Carlo integration techniques explained in [201].

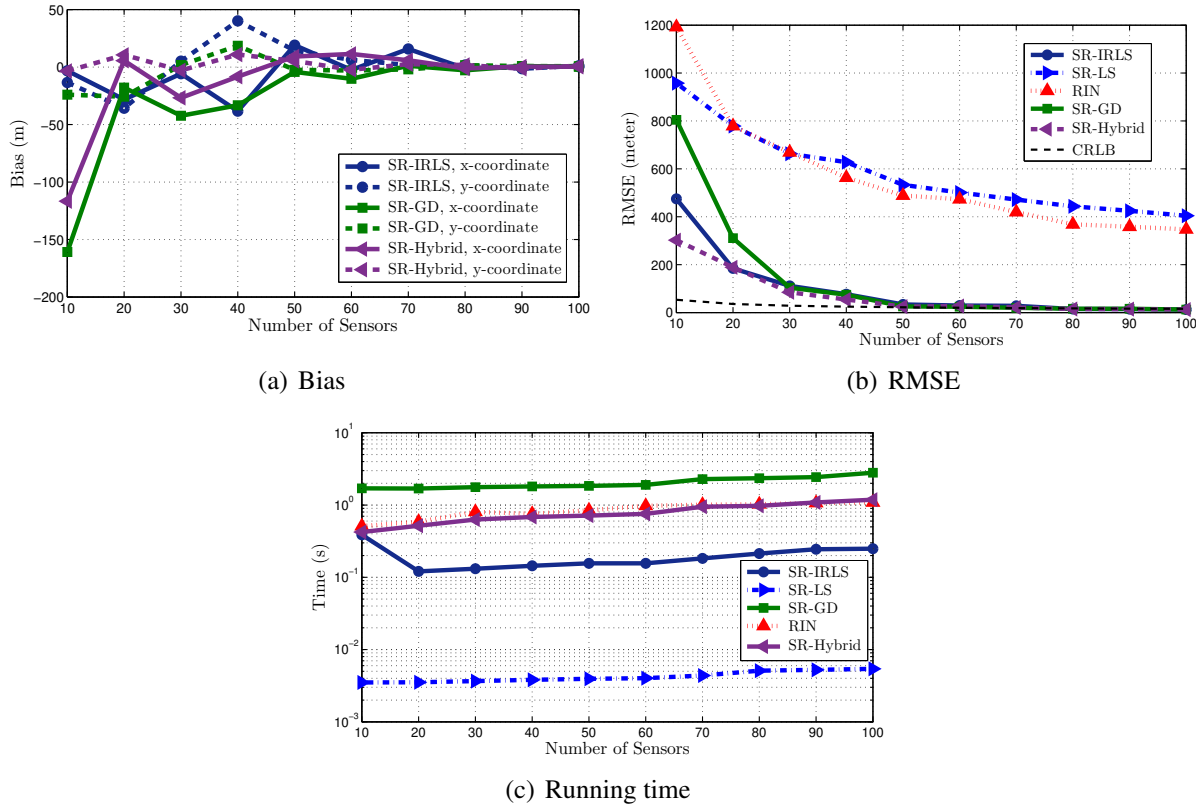


Figure 10.3: Performance of the localization methods versus number of sensors for 1000 Monte Carlo trials and $\beta = 0.4$.

Figure 10.3(c) shows the running times² for different number of sensors. Clearly, the iterative methods requires more computation time than the least square method. Also, as it was expected and can be noticed in Figure 10.1, the running time of the hybrid method is less than SR-GD, but more than SR-IRLS.

²Running time reflects the time required to execute all the steps of the algorithms, including initialization, preprocessing, convergence, and post-processing. All simulations have been performed under MATLAB 2014a environment on a PC equipped with Intel Xeon E5-1650 processor (3.20 GHz) and 8 GB of RAM.

It is also worthwhile to compare the performance of the localization methods for the case when no sensor is reporting unreliable measurements and the range measurements are corrupted only by an additive Gaussian noise, i.e. $\beta = 0$. As it can be seen in Figure 10.4, the LS method outperform the robust methods. This was expected since the LS methods are particularly tailored to deal with Gaussian noise, while the robust methods are customized to handle the unreliable measurements. We are sacrificing efficiency for $\beta = 0$, to achieve stability in deviation from the model. However, it is easy to notice that the RMSE of the proposed robust methods is close to the RMSE of the SR-LS method, which implies the near optimal performance for Gaussian noise.

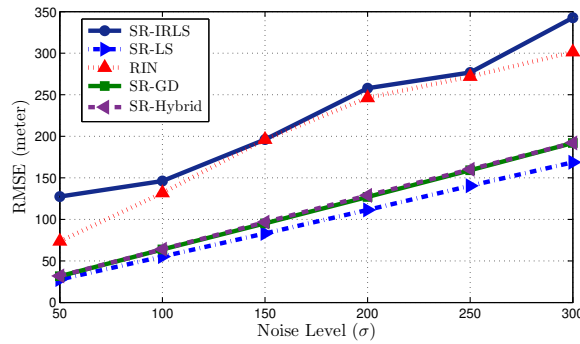


Figure 10.4: Comparison of the RMSEs in an environment with no outlier sensor, $\beta = 0$.

10.3.2 Scenario II

In this section, the problem of localizing a target in a radio cellular network is considered. The network consists $R = 8$ base stations (BSs), which are trying to estimate the location of a target in a city center area. The configuration of the BSs and the city center, as depicted in Figure 10.5, is taken from a realistic network [201].

The outlier-free measurements are result of line-of-sight (LOS) sensings. On the other hand NLOS sensings produce unreliable measurements. Field trials have indicated that the measurement errors

in harsh LOS/NLOS environments can be modeled as a Gaussian mixture distribution [201],

$$p_V(v) = (1 - \beta)\mathcal{N}(v; 0, \sigma^2) + \beta\mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2), \quad (10.20)$$

where $\mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2)$ is a Gaussian distribution with mean μ_{NL} and variance σ_{NL}^2 , modeling the NLOS measurements.

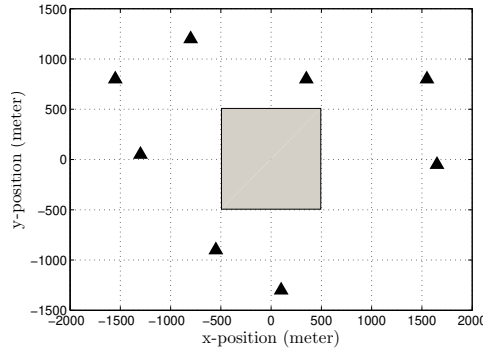


Figure 10.5: Geometry of the sensors, marked as triangle, and the city center area, marked as gray square, in a real world operating cellular radio network.

For each BS, we obtain K measurements and stack them in the measurement vector as follows:

$$\mathbf{b} = \begin{bmatrix} r_1(1)^2 - \|\mathbf{a}_1\|^2 \\ \vdots \\ r_1(K)^2 - \|\mathbf{a}_1\|^2 \\ \vdots \\ r_R(1)^2 - \|\mathbf{a}_R\|^2 \\ \vdots \\ r_R(K)^2 - \|\mathbf{a}_R\|^2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} -2\mathbf{a}_1^T & 1 \\ \vdots & \vdots \\ -2\mathbf{a}_1^T & 1 \\ \vdots & \vdots \\ -2\mathbf{a}_R^T & 1 \\ \vdots & \vdots \\ -2\mathbf{a}_R^T & 1 \end{bmatrix}. \quad (10.21)$$

In the simulations, it is assumed that each BS reports $K = 20$ samples. The measurement errors are drawn from the distribution in (10.20) with $\sigma = 55$, $\mu_{NL} = 380$, and $\sigma_{NL} = 120$. The position of the target is uniformly generated in the city center area.

Figure 10.6 illustrates the performance of different localization methods versus the contamination ratio for $0 \leq \beta \leq 1$. This figure shows that SR-GD outperforms its competitors. Moreover, the hybrid version and SR-IRLS perform the same in this configuration and are able to handle NLOS measurements up to a certain amount and meet the CRLB up to a certain β . For, Large values of β , the SR-IRLS method breaks down, but still works better than the least square method. However, in this scenario SR-GD is able to localize the target for even large contamination ratios.

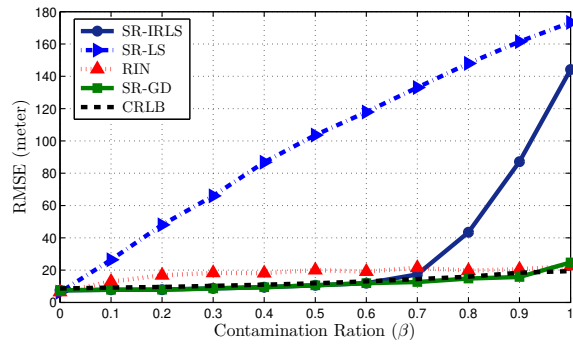


Figure 10.6: Mean RMSE of different localization methods versus contamination ratio, for 100 MC trials.

Moreover, the RIN method performs accurately in this scenario, in comparison with the previous scenario. In this scenario, the RIN can estimate the PDF of the error more accurately. This was expected because, firstly, we are collecting $R \times K = 160$ measurements, secondly, the measurement error has a mixture Gaussian distribution. As a result, the RIN method can produce a better estimate of the target location. With 160 measurements, RIN is able to approximate the measurement error distribution. Thus, the its RMSE does not change considerably for different values of β . This fact is vividly clear for the extreme case. For $\beta = 1$, the RIN method is able to approximate $\mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2)$ as the PDF of the measurement error. As a result, this method outperforms the competitors for the special case of $\beta = 1$.

10.4 Conclusions

In this chapter, we have considered the problem of localizing a single target in the presence of unreliable measurements with unknown probability distribution. For that, the squared-range formulation is exploited. To disregard the outlier measurements and find the estimate using the reliable measurements, we have used robust statistics. Then the problem is converted into a known class of optimization problems, namely GTRS, using the concepts in robust statistics. Two algorithms and a hybrid method are proposed to solve the problem. Convergence of the algorithms is analyzed theoretically.

The simulation results suggest that the proposed methods outperform the existing methods, while providing a near optimal performance for Gaussian noise.

CHAPTER 11: CONCLUSION

11.1 Contributions

In this dissertation, we discussed the problem of robust and scalable data representation and analysis as three major tasks. In the first task, supervised representation learning, we mainly focused on deep neural networks, which have been proven to be very efficient in finding abstract and useful representations of the data whenever huge amount of data are available. Deep neural networks consist of composition of many nonlinear transformations and are notoriously difficult to train, as they become deeper. We studied the optimization dynamics of a certain family of deep neural networks and illustrated how such dynamics can be improved by regularizing the singular values of the linear operators in the network. We also showed how we can robustify the learned representations to outliers, by enforcing certain structures on the embedding space learned by the neural network. Next, we showed how we can employ the feature space generated by neural networks to select a representative subset of our data set.

On the other hand, in the scenarios when a large amount of data is not available, specific domain knowledge can be employed to find useful representations in an unsupervised manner. Thus, for the second task, unsupervised representation learning, we discussed methods that do not need labelled data to reveal the latent structures. For example, compressive sensing (CS) states that the compressed representations of the signal, achieved by simple random projections, contain enough information to reconstruct the original signal, as long as it has a sparse representation in any sparsifying basis. We showed that how we can use Bayesian inference to incorporate our prior knowledge to obtain better random projections and to improve the reconstruction accuracy. We also showed that we can use sparsity constraint to enforce structures, such as piece-wise continuity, on the latent factors of data. However, optimization with sparsity constraint is not a trivial task. Thus,

we proposed an optimization technique, employing Majorization-Minimization, to guarantee the convergence.

For the third task, we examined how we can make a data processing system fault-tolerant both in the data collection and the data processing phases. Due to recent advancements in the production of inexpensive data processing and collection devices, many of the experiments in machine learning consist of processing thousands of samples over thousands of processing nodes. But oftentimes the cost function of optimization problems does not lend itself to asynchronous parallel computation, which makes the framework sensitive to faulty processors. Thus, we proposed a new framework to solve the CS recovery problem in an asynchronous manner. We also showed how sparsity constraint can be used during optimization to disregard the outlier measurements. For that, we study a certain family of optimization problems, namely Generalized Trust Region Subproblem, and show how we can robustify it with a provably convergent approach.

For each method, we showed the effectiveness of the proposed approach in real-world problems such image/video classification, active learning, dataset summarization, video summarization, face image retrieval, and wireless sensor networks. Here, we summarize the contributions of this dissertation in more details:

11.1.1 Analyzing and Improving the Optimization Dynamics of Deep Residual Networks

We analyzed the optimization dynamics of Residual Networks [12] and showed how adding skip connections to the network helps with the optimization stability. Specifically, we proved that the norm of the gradient vector is preserved, as it is propagated through the layers via the chain rule. We derived bounds for the norm-preservation of the neural network and showed that residual blocks become more norm-preserving as the number of block increases. Our theoretical analysis was verified by extensive empirical studies. We also proposed a new regularization to improve

this norm-preservation behaviour in the network. Such regularization leads to more stable optimization and better performance. Our proposed scheme is based on setting the singular values of the convolution operator such that on average the norm of the gradient vector is preserved under the backpropagation. We showed that, by employing our proposed regularization, we can achieve same classification accuracy by using up to 4 times smaller networks.

11.1.2 Union of Low-Dimensional Subspace for Outlier Detection Using Deep Neural Networks

We showed that by engineering the embedding space generated by a deep neural network, we can make it more robust to the outlier samples. It has been shown that the samples embedded into a feature space generated by a conventional softmax classifier follow a mixture of Gaussian distributions [19]. However, such embeddings do not necessarily make the outlier samples more distinguishable. We also know that, due to the sheer representation ability of neural networks, they can generate embedding spaces with different pre-defined structures. Thus, we proposed to embed the samples from each class onto a 1-dimensional subspace in the embedding space. We showed that enforcing such structure helps us to detect samples that do not belong to any of the classes/subspaces with higher accuracy, compared to the conventional class-condition Gaussian distribution. Our proposed scheme can be easily applied to different architectures and different data modalities. For example, in [51], we showed how we can use our proposed scheme in both image and video classification tasks.

11.1.3 Design and Analysis of a Linear-Time Subset Selection Algorithm for Large-Scale Data

We proposed and analyzed a new subset selection algorithm, whose time complexity increases linearly with the number of samples. This is an important property as the size of the datasets is growing larger and larger. Thus, the task of selecting the most representative or the most informa-

tive subset of data in tractable time is becoming more difficult. We showed that our approach can produce results in tractable time for datasets containing more than a million samples. We showed the effectiveness of our proposed algorithm in different tasks such active learning for human action recognition, dataset summarization, and video summarization.

11.1.4 Design and Analysis of Provably Convergent Optimizers for Problems with Sparsity Constraint

We showed how we can use sparsity constraint to both reveal latent structures in the data and how to make an optimization problem more robust to faulty data points. For instance, we used sparsity to obtain piece-wise continuous latent factors in a non-negative matrix factorization problem. We also derived a provably convergent optimizer for the proposed problem and showed how we can use our scheme in a wireless sensor network to separate the signals from different transmitters. As another example, we used sparsity constraint to disregard the faulty measurements in a source localization problem. For that, we proposed an optimization problem that looks for a solution that satisfies as many measurements as possible, instead of considering all the measurements. We also proposed an algorithm to solve this problem in a provably convergent manner. We showed that our proposed scheme can meet the Cràmer-Rao lower bound for a sufficiently large number of measurements.

11.1.5 Bayesian Inference for Efficient and Robust Compressive Sensing

We studied the application of graphical models in both the sampling and reconstruction steps of Compressive Sensing. For the sampling phase, we proposed to employ our prior knowledge of slow variation in the signal over time to distribute the sampling energy more efficiently over the signal components. For that, we designed a graphical model that can assign importance scores to each

signal component. Therefore, we can reconstruct the more important parts of the signal with higher accuracy, at the expense of losing some accuracy on the less important parts of the signal. For the reconstruction phase, we used a similar graphical model to solve the Compressive Sensing recovery problem in an asynchronous parallel manner. This means that our proposed recovery scheme can tolerate existence of slow and faulty processors among the processing nodes. To achieve this, we designed a graphical model that is able to assign reliability scores to the processors and share the information among processors, while considering the reliability scores and without waiting for the slower ones. We analyzed the convergence of our proposed Bayesian inference component both theoretically and empirically.

11.2 Future Directions

We presented an array of supervised and unsupervised techniques to improve the scalability and robustness of the data processing systems. The current state-of-the-art supervised techniques have a few drawbacks. First, they are very data hungry. This makes them expensive both in terms of data collection and computation time. One possible solution that has been gaining momentum recently is to reduce the data collection cost by allowing some noise in the data. It is a well-known fact that the recent success of the neural networks is due to the emergence of large-scale datasets. Deep neural networks often require huge, and sometimes impractical, amount of data to perform well. However, crowd sourcing platforms and data collection bots have provided the opportunity to collect a large number of samples in a very cost-efficient manner. Such approaches inevitably introduce some error during the labelling or annotation process. Such mislabeled samples can easily cause severe performance degradation for any classifier, and specially for deep neural networks. It has been shown that the deep neural networks have the representational ability to memorize data points with completely random labels [215], which emphasizes their sensitivity to label noise and

the need for robustifying techniques. A robust deep neural network provides us with the opportunity of using large amounts of inexpensive noisy data to achieve very high classification accuracy. In Chapter 4, we discussed techniques to detect outliers samples at testing time, also known as out-of-distribution detection. Similar ideas can be employed to prune the training set and to detect outliers at training time. For example, the proposed spectral discrepancy measure can be used to assign weights to the training sample and de-emphasize the outlier samples. It is also worthwhile to mention that enforcing the union of 1-dimensional structure, discussed in Chapter 4, does not necessarily require labeled data. Thus, it is possible to constrain the feature vectors in the embedding space employing a large number of unlabelled data points and a few labelled samples.

Another drawback of current learning systems is lack of theoretical guarantees for optimization success and the generalization error. It is still not clear why and how very deep neural networks with millions of parameters are able to converge to a solution to a non-convex problem using fairly simple optimization techniques. In Chapter 3, we tried to provide some insight for a specific case, but more theoretical insight to this problem is very much needed. The regularization framework proposed in this chapter provided us with some insight to the inner workings of residual networks, but faster, more efficient, techniques are needed to make such regularization more practical. The main computational cost of our proposed regularization is the calculation of all the singular values of the convolution operator. Calculating or approximating the singular values in a computationally lightweight manner can lead to a new wave of regularization methods in deep neural network literature. Such methods can be used to control how the neural networks preserve the distance between data points or how smooth they warp the input space to create the feature space. This is an important property to study and regularize, as it has been shown that the neural network can be easily fooled by adding a very small, barely visible, noise to the input sample. Thus, having some control over singular values can help us to detect outliers and to robustify the system against adversarial attacks.

**APPENDIX A: PROOFS FOR THEORETICAL RESULTS PRESENTED
IN CHAPTER 3**

A.1 Proof of Theorem 3.1

For a cost function $\mathcal{E}(\cdot)$ and the Jacobian \mathbf{J} of \mathbf{x}_{l+1} with respect to \mathbf{x}_l , applying chain rule, following is true:

$$\begin{aligned}\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} &= \mathbf{J} \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}}, \\ \mathbf{J} &= \frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_l} = \mathbf{I} + DF_l(\mathbf{x}_l),\end{aligned}$$

where D is the differential operator and for any \mathbf{v} with bounded norm we have:

$$DF_l(\mathbf{x}_l)\mathbf{v} = \lim_{t \rightarrow 0^+} \frac{F_l(\mathbf{x}_l + t\mathbf{v}) - F_l(\mathbf{x}_l)}{t}$$

To prove Theorem 3.1, we first state a lemma.

Lemma A.1. *For any non-singular matrix $\mathbf{I} + \mathbf{M}$, we have:*

$$1 - \sigma_{\max}(\mathbf{M}) \leq \sigma_{\min}(\mathbf{I} + \mathbf{M}) \leq \sigma_{\max}(\mathbf{I} + \mathbf{M}) \leq 1 + \sigma_{\max}(\mathbf{M}),$$

where $\sigma_{\min}(\mathbf{M})$ and $\sigma_{\max}(\mathbf{M})$ represent the minimum and maximum singular values of \mathbf{M} , respectively.

Proof. Since $\sigma_{\min}(\mathbf{I} + \mathbf{M}) > 0$, the lower bound is trivial for $\sigma_{\max}(\mathbf{M}) \geq 1$. For $\sigma_{\max}(\mathbf{M}) < 1$, it is known that $|\lambda_{\max}(\mathbf{M})| < 1$, where $\lambda_{\max}(\mathbf{M})$ is the maximum eigenvalue of \mathbf{M} [216]. Thus, we can show that:

$$\begin{aligned}\sigma_{\min}(\mathbf{I} + \mathbf{M}) &= (\sigma_{\max}((\mathbf{I} + \mathbf{M})^{-1}))^{-1} = \|(\mathbf{I} + \mathbf{M})^{-1}\|_2^{-1} \stackrel{(a)}{=} \left\| \sum_{k=0}^{\infty} (-1)^k \mathbf{M}^k \right\|_2^{-1} \\ &\geq \left(\sum_{k=0}^{\infty} \|(-1)^k \mathbf{M}^k\|_2 \right)^{-1} \geq \left(\sum_{k=0}^{\infty} \|\mathbf{M}\|_2^k \right)^{-1} = \left(\frac{1}{1 - \|\mathbf{M}\|_2} \right)^{-1} = 1 - \sigma_{\max}(\mathbf{M}).\end{aligned}$$

Identity (a) is known as Neuman series of a matrix, which holds when $|\lambda_{\max}(\mathbf{M})| < 1$ and $\|\cdot\|_2$ represents the l_2 -norm of a matrix.

The upper bound is easier to show. Due to triangle inequality:

$$\sigma_{\max}(\mathbf{I} + \mathbf{M}) = \|\mathbf{I} + \mathbf{M}\|_2 \leq \|\mathbf{I}\|_2 + \|\mathbf{M}\|_2 = 1 + \sigma_{\max}(\mathbf{M}).$$

□

Thus, knowing that

$$\sigma_{\min}(\mathbf{J}) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \mathbf{J} \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \sigma_{\max}(\mathbf{J}) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2,$$

using Lemma A.1, we conclude that

$$(1 - \delta') \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} \right\|_2 \leq (1 + \delta') \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2.$$

where, $\delta' = \sigma_{\max}(DF_l(\mathbf{x}_l))$. Furthermore, we have:

$$\begin{aligned} \sigma_{\max}(DF_l(\mathbf{x}_l)) &= \sup_{\mathbf{v}} \frac{\|DF_l(\mathbf{x}_l)\mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sup_{\mathbf{v}} \frac{1}{\|\mathbf{v}\|_2} \left\| \lim_{t \rightarrow 0^+} \frac{F_l(\mathbf{x}_l + t\mathbf{v}) - F_l(\mathbf{x}_l)}{t} \right\|_2 \\ &= \lim_{t \rightarrow 0^+} \sup_{\mathbf{v}} \frac{1}{\|\mathbf{v}\|_2} \left\| \frac{F_l(\mathbf{x}_l + t\mathbf{v}) - F_l(\mathbf{x}_l)}{t} \right\|_2 = \lim_{t \rightarrow 0^+} \sup_{\mathbf{v}} \frac{\|F_l(\mathbf{x}_l + t\mathbf{v}) - F_l(\mathbf{x}_l)\|_2}{t\|\mathbf{v}\|_2} \\ &\leq \|F_l\|_L, \end{aligned}$$

where $\|f\|_L$ is the Lipschitz seminorm of function f and is defined as

$$\|f\|_L := \sup_{\mathbf{x} \neq \mathbf{y}} \frac{\|f(\mathbf{x}) - f(\mathbf{y})\|_2}{\|\mathbf{x} - \mathbf{y}\|_2}.$$

To conclude the proof, we use the following lemma:

Lemma A.2. (Theorem 1 in [40]) Suppose we want to represent a nonlinear mapping $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$, satisfying Assumption 3.1, with a sequence of L non-linear residual blocks of form $\mathbf{x}_{l+1} =$

$\mathbf{x}_l + F_l(\mathbf{x}_l)$. There exists a solution such that for all residual blocks we have $\|F_l\|_L \leq c \frac{\log(2L)}{L}$.

Therefore, $\delta' = \sigma_{\max}(DF_l(\mathbf{x}_l)) \leq \|F_l\|_L \leq c \frac{\log(2L)}{L} = \delta$, which concludes the proof.

A.2 Proof of Theorem 3.2

In the classical back-propagation equation, for a cost function $\mathcal{E}(\cdot)$ and the Jacobian \mathbf{J} of \mathbf{x}_{l+1} with respect to \mathbf{x}_l , applying chain rule, following is true:

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} &= \mathbf{J} \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}}, \\ \mathbf{J} &= \frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_l} = \mathbf{I} + \mathbf{W}_l^T, \end{aligned} \tag{A.1}$$

To prove the theorem, using Lemma A.1 and knowing that

$$\sigma_{\min}(\mathbf{J}) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \mathbf{J} \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \sigma_{\max}(\mathbf{J}) \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2,$$

we conclude that

$$(1 - \delta') \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2 \leq \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} \right\|_2 \leq (1 + \delta') \left\| \frac{\partial \mathcal{E}}{\partial \mathbf{x}_{l+1}} \right\|_2.$$

where, $\delta' = \sigma_{\max}(\mathbf{W}_l)$. To conclude the proof, we use the following lemma.

Lemma A.3. (Theorem 2.1 in [32]) Suppose $L \geq 3\gamma$. Then, there exists a global optimum for $\mathcal{E}(\mathcal{W})$, such that we have

$$\sigma_{\max}(\mathbf{W}_l) \leq \frac{2(\sqrt{\pi} + \sqrt{3}\gamma)^2}{L}, \forall l = 1, 2, \dots, L,$$

where γ is $\max(|\log \sigma_{\max}(\mathbf{R})|, |\log \sigma_{\min}(\mathbf{R})|)$.

Using the results from this lemma and setting $\delta = \frac{2(\sqrt{\pi} + \sqrt{3\gamma})^2}{L}$, Theorem 3.2 follows immediately.

A.3 Proof for Corollary 3.1

Here, Jacobian matrix is $\mathbf{J} = \mathbf{I} + \mathbf{F}'\mathbf{W}^{(1)T}\mathbf{F}'\mathbf{W}_l^{(2)T}$, where \mathbf{F}' is the Jacobian of $\rho(\cdot)$ with respect to its input. Since we know that $0 \leq \frac{\partial \rho_n(\mathbf{x})}{\partial x_{n'}} \leq c_\rho, \forall n = n'$ and $\frac{\partial \rho_n(\mathbf{x})}{\partial x_{n'}} = 0, \forall n \neq n'$, we have $\|\mathbf{F}'\|_2 \leq c_\rho$. Therefore:

$$\|\mathbf{F}'\mathbf{W}_l^{(1)T}\mathbf{F}'\mathbf{W}_l^{(2)T}\|_2 \leq \|\mathbf{F}'\|_2\|\mathbf{W}_l^{(1)T}\|_2\|\mathbf{F}'\|_2\|\mathbf{W}_l^{(1)T}\|_2 \leq c_\rho^2\|\mathbf{W}_l^{(1)}\|_2\|\mathbf{W}_l^{(2)}\|_2$$

and using Lemma A.1 and setting $\delta = c_\rho^2\|\mathbf{W}_l^{(1)}\|_2\|\mathbf{W}_l^{(2)}\|_2$, Corollary 3.1 follows immediately.

**APPENDIX B: IMPLEMENTATION DETAILS AND FURTHER
EXPERIMENTAL RESULTS FOR THE METHOD PROPOSED IN
CHAPTER 4**

B.1 Implementation Details

For the image classification task, we deploy WideResNet with depth 28 and width 10 as the neural network architecture for our method. All the network parameters are set as the original implementation in [77], except the last layer which is modified as proposed in Chapter 4. Stochastic gradient descent (SGD) with momentum of 0.9 is used to train the network for 200 epochs with batch size of 128. At the beginning of the training, the learning rate is set to 0.1 and it is then dropped by a factor of 10 at 50% and 75% of the progress. Weight decay is set to 5×10^{-4} . At the test time, we draw 50 Monte Carlo samples to estimate $p(\phi_n \leq \phi^*)$ and to detect the OOD samples. To enforce the structure, the last fully-connected layer is initialized with *orthonormal* weights, using the method discussed in [72]. Then, to assign class membership probabilities, softmax function is used on the cosine similarities between the feature vector and the rows the fully-connected layer using $p_{ln} = \frac{e^{|\cos(\theta_{ln})|}}{\sum_l e^{|\cos(\theta_{ln})|}}$. Algorithm 8 summarizes the training and testing phases of the proposed approach.

Algorithm 8 OOD detection using Union of 1D Subspaces.

Input: ID training dataset, testing set, critical spectral discrepancy ϕ^* , Number of Monte Carlo samples S

Training:

Interclass constraint: Freeze weights in the last FC layer such that $\mathbf{w}_l^T \mathbf{w}_{l'} = 0, l \neq l', \forall l, l' = 1, \dots, L$
Intraclass constraint: use (4.1) as the loss function

Testing:

- 1: Compute $v_1^{(l)}$ for each class l using training feature vectors
 - 2: **for** i_n in the testing set **do**
 - 3: Sample S feature vectors $\mathbf{x}_n^s, s = 1, \dots, S$
 - 4: Compute ϕ_n^s for each sample \mathbf{x}_n^s using (4.2)
 - 5: Estimate $p(\phi_n \leq \phi^*)$ using (4.3)
 - 6: **if** $p(\phi_n \leq \phi^*) = 0$ **then**
 - 7: Classify i_n as an OOD sample
 - 8: **else**
 - 9: Use $p_{ln} = \frac{e^{|\cos(\theta_{ln})|}}{\sum_l e^{|\cos(\theta_{ln})|}}$ to assign class membership
 - 10: **end if**
 - 11: **end for**
-

B.2 Additional Experiments

Here, we report additional experimental results. The dataset and the evaluation metrics are the same as Chapter 4.

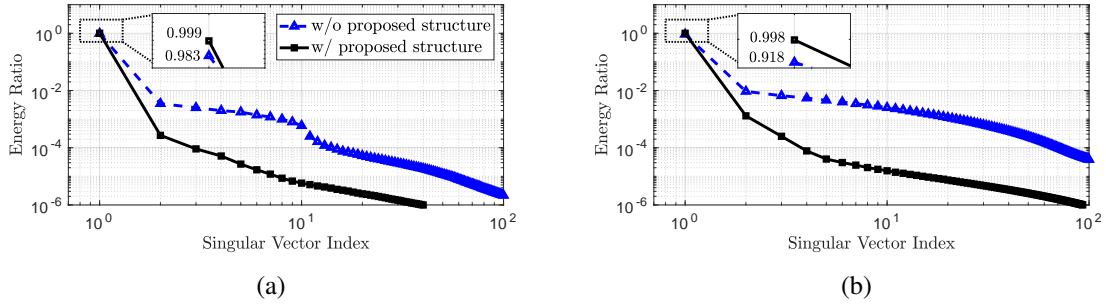


Figure B.1: Energy Ratio of the training samples along the first 100 singular vectors of features extracted using WideResNet with and without our proposed embedding trained on (a) CIFAR10 and (b) CIFAR100. The proposed embedding increases the energy along the first singular vector from 98.3% to 99.9% for CIFAR 10 and from 91.8% to 99.8% for CIFAR100.

Figure B.1 demonstrates the impact of the proposed training scheme on the spectrum of the feature vectors. This figure shows the ratio of the energy concentrated along each singular vector averaged over all the classes. The energy ratio along the i^{th} singular vector is calculated as $\frac{\lambda_i}{\sum_j \lambda_j}$. As discussed in Section 4.2, our goal is to make the feature vectors of each class to lie on a 1-dimensional subspace and to make the gap between the first eigenvalue λ_1 and other eigenvalues $\lambda_j, j > 1$ as large as possible. Figure B.1 illustrates that the proposed training scheme can effectively achieve this by increasing the energy ratio along the first singular vector and reducing the energy concentrated along the rest of the singular vectors. Consequently, the first singular vector of each class will be more robust to noise.

Figure B.2 demonstrates the robustness of the first singular vector to outliers in a toy scenario. For this experiment, the feature vectors from a single class of CIFAR10 are extracted using the network

trained with our proposed structure. Then, some percentage of the vectors are replaced by feature vectors from the other classes, which act as outliers. The figure shows the correlation between the singular vectors of contaminated and clean data for different noise levels, averaged over 10 trials. Correlation of 1 means that the direction of the singular vector has not changed at all after the introduction of the outliers. This experiments illustrate the fact that the first singular vector of the data is very robust to outliers and its direction does not change much even after replacing about half of the samples. This experiment validates the motivation behind our method, which is the robustness of the first singular vector. It is worthwhile to mention that, in OOD detection setting studied in this work, we do not have such severe contamination, as $v_1^{(l)}$ is extracted from the training set and only a small subset of the feature vectors might be noisy due to training error or misclassification.

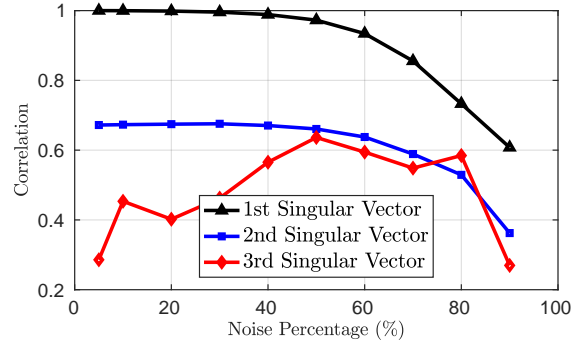


Figure B.2: Correlation of different singular vectors of noisy data with the same singular vector of clean data, averaged over 10 trials. Feature vectors corresponding to the first class of CIFAR10 act as the data and the feature vectors belonging to other classes are used as outliers. Noise levels up to 50% have almost no impact on the direction of the first singular vector.

Figure B.3 examines the number of Monte Carlo samples necessary for a good estimation of $p(\phi_n < \phi^*)$. It shows that having as low as 10 samples can improve the results. However, as expected, having more samples always leads to better estimation and better performance. It is also worthwhile to mention that since the samples can be drawn concurrently, drawing more samples

does not increase the running time much.

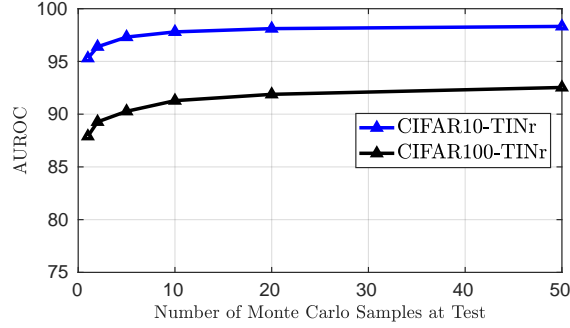


Figure B.3: Area Under ROC curve using the proposed framework versus the number of the Monte Carlo samples used for estimating $p(\phi_n < \phi^*)$. The networks are trained on CIFAR10 and CIFAR100 and tested on TINr as the OOD dataset.

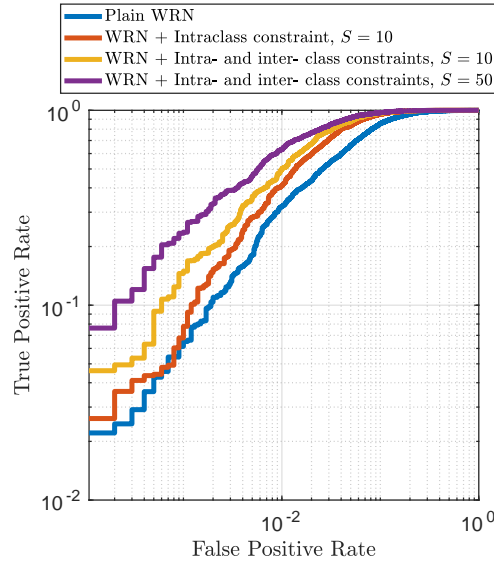


Figure B.4: ROC curves for different variants of the proposed scheme in logarithmic scale, using CIFAR10 (ID) and TINr (OOD). WideResNet (WRN) with depth of 28 and width of 10 is used as the deep feature extractor.

Figure B.4 shows the true positive rate against false positive rate, also known as the receiver operating characteristic (ROC) curve, for different variants of the proposed architecture. This figure

demonstrates how each component of the method, such as intraclass constraint, interclass constraint, and number of Monte Carlo (MC) samples S , affect the ROC.

In this study, CIFAR10 is used as the in-distribution (ID) dataset and the resized version of the TinyImagenet (TINr) is used as the out-of-distribution (OOD) dataset. $p(\phi_n < \phi^*)$ is used for OOD detection in all the different variants, even for the baseline, i.e., Plain WideResnet. However, no MC sampling is performed for the baseline architecture. Specifically, enforcing only the intraclass constraint on the model and using only $S = 10$ MC samples increases the area under the ROC curve (AUROC) by about 3%, from 94.7% to 96.3%. On the other hand, imposing the interclass constraint, i.e., enforcing *orthogonality* on the subspaces, improve the AUROC by another 1%. Finally, as expected, using more samples to estimate $p(\phi_n < \phi^*)$ can also increase the AUROC. For example, increasing the number of samples from 10 to 50 can improve the results by another 1.2%, leading to AUROC of 98.5%.

Table B.1: Detection errors and f1-scores achieved by setting ϕ^* using the training set, compared to the best achievable values, on different pairs of ID and OOD datasets.

Training dataset	OOD dataset	Detection Error		F1 Score	
		Fixed ϕ^*	Best ϕ^*	Fixed ϕ^*	Best ϕ^*
CIFAR10	TINc	10.4	6.8	90.5	93.0
	TINr	7.6	6.2	92.5	93.6
	LSUNc	8.6	3.7	93.1	96.2
	LSUNr	4.1	3.8	95.0	96.1
CIFAR100	TINc	19.8	18.9	79.2	81.0
	TINr	17.6	14.2	83.2	86.0
	LSUNc	14.9	13.9	76.1	76.9
	LSUNr	12.9	11.3	85.3	88.6

AUROC, as well as the area under the precision-recall curve (AUPR) and false positive rate at true positive rate of 95% that are reported in Chapter 4, is independent of the value of the critical spectral discrepancy ϕ^* . However, f1-score and detection error do depend on ϕ^* . In Chapter 4,

we reported the best detection errors and f1-scores achievable by the baselines and our proposed method. Here, we investigate the impact of choosing ϕ^* using the training set. In general, having ϕ^* as a parameter gives us the freedom to tune the precision and recall according to the requirements of the application at hand. To fix ϕ^* using the training set, we choose a value for which most, say 98%, of the training samples have a spectral discrepancy of less than this value. Table B.1 summarizes the results and compares them with the best achievable detection errors and f1-scores. It is evident that the results are not far from the best achievable results. This indicates that the training set can be used to set the value of ϕ^* or to estimate the general proximity of best ϕ^* .

Table B.2: Performance of different OOD detection tests, in term of AUROC, for distinguishing ID and OOD test set data.

Training dataset	OOD dataset	OOD Test		
		$p(\phi_n \leq \phi^*)$	$\mathbb{E}\{\phi_n\} \leq \phi^*$	$\phi_n \leq \phi^*$
CIFAR10	TINc	98.1	95.8	95.6
	TINr	98.5	95.5	95.6
	LSUNc	99.4	96.5	96.9
	LSUNr	99.3	96.6	96.4
CIFAR100	TINc	89.1	87.0	85.7
	TINr	93.7	85.9	85.2
	LSUNc	93.8	88.0	87.0
	LSUNr	95.7	93.0	91.2

Finally, Table B.2 compares the results, in terms of AUROC, for different OOD detection tests. Motivated by our theoretical investigation in Section 4.2, we proposed to use $p(\phi_n \leq \phi^*)$ for OOD detection. This is because if the feature vectors belonging to the known classes lie on 1-dimensional subspaces, the OOD feature vectors will occupy the same region with probability 0, unless they are drawn from the exact same distribution. Here, we demonstrate the results for a few more OOD detection tests. In particular, expected spectral discrepancy of each sample $\mathbb{E}\{\phi_n\}$ can also be used for OOD detection. $\mathbb{E}\{\phi_n\}$ can be estimated using a similar MC sampling technique. Furthermore, one can perform a single conventional forward pass and calculate a point

estimate of ϕ_n . This table shows the performance for each of these tests. This table confirms our theoretical investigation and shows that using $p(\phi_n \leq \phi^*)$ is the most accurate OOD test. This is partly because ID test samples might have an expected spectral discrepancy outside the tiny region occupied by ID training samples, but they will have a nonzero probability in that region. While on the hand, the OOD samples will rarely have nonzero probability inside the same region. This also shows that addition of MC sampling and using probabilistic OOD tests, such as expected value, is not enough for good detection performance. The OOD detection test needs to reflect the underlying structure of data in the feature space and co-design of the embedding function and the OOD test can lead to significant improvements.

Method	Extra Information Used
Discrepancy Loss [59]	OOD samples during training
Outlier Exposure [58]	OOD samples during training
Word Embedding [64]	Auxiliary text data to achieve better embedding during training
ODIN [14]	OOD samples for validation (to tune perturbation magnitudes for adversarial examples)
Mahalanobis [19]	OOD samples for validation (to tune hyperparameters or perturbation magnitudes for adversarial examples)
GPND [57]	OOD samples for validation (to tune penalty terms and latent space size)
Confidence Loss [62]	OOD samples for validation (to tune penalty term)
Likelihood Ratio [63]	OOD samples for validation (to tune hyperparameter μ)
Ensemble [15]	OOD samples for validation (hyper parameter tuning)
OLTR [66]	None (but is able to leverage OOD samples for validation)
Softmax Pred. [13]	None
Conterfactual [65]	None
Generalized ODIN [67]	None
CROSR [18]	None

Table B.3: A non-exhaustive summary of recent OOD detection methods. The information provided in the table is extracted from their corresponding manuscript or the code provided by authors.

B.3 Related Methods: Leveraging OOD Data for OOD Detection

In scenarios where a subset of OOD samples is available at the training time, they can be used to improve the performance. Authors in [58, 59] have shown the advantage of the using OOD samples during training. The main idea is to create a feature space such that the ID samples are as distinguishable as possible from OOD samples, by maximizing the distance between the ID

samples and OOD samples. Other modalities of data, such as text, can also be leveraged to obtain a better embedding [64].

However, most of OOD detection methods make the assumption that OOD samples are not available during training, but a very small subset is available to tune some of the hyperparameters. For instance, ODIN [14] uses perturbed test samples and temperature scaling to reject the samples that are less robust to perturbations. OOD samples are used to tune the magnitude of the adversarial perturbation. The method proposed in [19] is more related to our proposed approach. In [19], the Mahalanobis distance between the test feature vectors and the training ID samples is used to detect OOD samples. Similar to ODIN, the method in [19] uses OOD samples to find the best values for their proposed OOD classifier. For the scenario where adversarial examples are used to tune the hyperparameters, a subset of OOD samples is used to find the best magnitude of the adversarial perturbation. Similarly, [63] adds perturbations, which needs to be tuned using OOD samples, to the input samples and uses the likelihood ratio to detect OOD samples. Furthermore, there are many methods that do not use off-the-shelf classifiers and train new classifiers, autoencoders, or generative models to enforce their desired structure on the feature space. While most of the hyperparameters can be tuned using ID validation set, some of the hyperparameters such as the latent space size, loss terms, and regularization terms need to be tuned by OOD samples[15, 57, 62]. For instance, [15] exploits OOD samples for early stopping of the ensemble of the classifiers, as well as hyperparameter tuning, and [57] uses them to find the best latent space size and penalty terms for the loss functions.

A few OOD detection methods rely only on ID validation set to tune hyperparameters. For example, the approach in [13] uses the softmax output to discriminate between the OOD and ID samples and, similar to our method, does not have any hyperparameters to be tuned by OOD validation set. Open Long-Tailed Recognition (OLTR) [66] creates a meta-embedding and employ the similarity to the known classes to reject OOD samples. Authors in [66] have shown that their method is

able to perform well with and without using OOD samples for hyper-parameter tuning. Similarly, methods in [18, 65, 67] only use ID validation set to tune the parameters of their model. Table B.3 provides a non-exhaustive summary of prior work and if/how they use extra information during training and validation phases.

**APPENDIX C: FURTHER EXPERIMENTAL RESULTS FOR THE
METHOD PROPOSED IN CHAPTER 5**

In this chapter, more experiments are provided to further investigate the performance of the proposed approach and to support the arguments presented in chapter 5. The implementation details are the same as in chapter 5, unless otherwise noted.

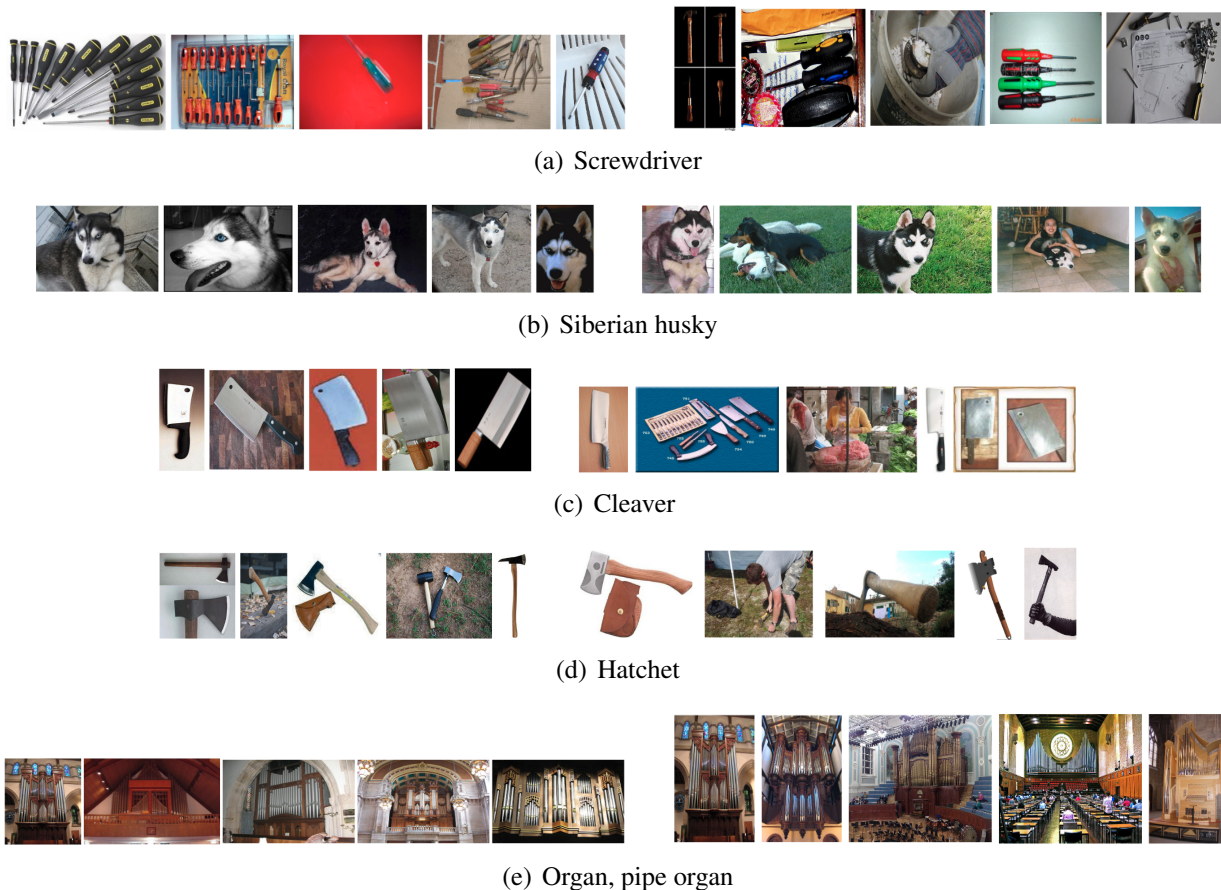


Figure C.1: Selected images by IPM (left) and K-medoids (right) from five sample classes of ImageNet [5]. Note that the IPM-selected samples are less cluttered with other objects, making them better representatives of the class.

C.1 Finding Representatives for ImageNet Dataset

Figure C.1 shows the selected samples using IPM and K-medoids from different classes of the ImageNet training set. DS3 and SMRS are too computationally expensive and do not generate

results for ImageNet in a tractable time. In this experiment, 5 images are selected as the representatives from each class. The implementation details are the same as given in Section 5.2.2.3. For each class, left row shows the images selected by IPM and right row shows the images selected by K-medoids. IPM-selected images are sorted by the order of selection, left-most sample being the first selected sample. Images selected by IPM are less cluttered with other objects and more representative of their corresponding classes. This leads to better classification accuracy, when the IPM-reduced representatives are used as the only labeled data available. This is demonstrated and discussed in Table 5.4. On the other hand, K-medoids, and other diversity-based selection methods, may select outliers or samples that may not be useful for classification task.

C.2 Finding Representatives for UCF-101 Dataset

Table C.1: Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.

Samples per class	1	2	3	4	5	6	7	8	9	10
Random	54.6	64.7	69.2	70.5	72.9	74.0	76.0	75.6	76.0	77.0
K-medoids	61.0	67.7	69.4	70.9	71.7	72.0	72.5	75.2	73.6	73.5
DS3[2]	60.8	69.1	74.0	75.2	74.9	75.3	75.8	77.0	77.6	76.6
IPM	65.3	72.6	74.9	77.6	77.0	78.5	78.4	78.4	79.0	78.2

Table C.1 shows the classification accuracy of ResNet18 trained using the representatives selected by different methods (extended version of Table 5.3). We compare IPM with DS3[2], K-medoids, and random selection as the baseline. To achieve accuracy of 77%, the closest competitor, i.e. DS3, requires 8 samples per class, while IPM achieves the same accuracy using half of that data. IPM adds the samples that contain the most information about the previously unseen space. This is because it selects the samples that are maximally correlated with the null space of currently selected samples. In contrast, methods such as K-medoids, that do not consider the current selected samples

fail to find the most critical samples, as we collect more samples.

This can be illustrated by t-SNE [3] visualization of the selection process. Figure C.2(Left) shows the 2D embedding of the points and the decision function learned by an SVM of different randomly selected pairs of UCF-101 dataset. On the right, the decision function learned by the same classifier, trained only on a few representatives, is demonstrated. This experiment demonstrates the fact that the representatives selected by IPM contain more information about the structure of the data. Compared to other selection methods and using the same number of samples, decision function learned by the classifier trained on the IPM-selected samples looks more similar to the decision function learned from all the data. This results in more accurate classification, as reported in Table C.1.

For a more qualitative investigation, Figure C.3 shows frames from the first selected representative by IPM (top row) and DS3 (bottom row) for a few classes of UCF-101 dataset. In this experiment, the first selected representative by K-medoids is the same as DS3 for all the classes. In general, in the clip selected by IPM, the critical features of the action, such as barbell, violin, kayak, and bow, are more visible and/or the bounding box for the action is bigger.

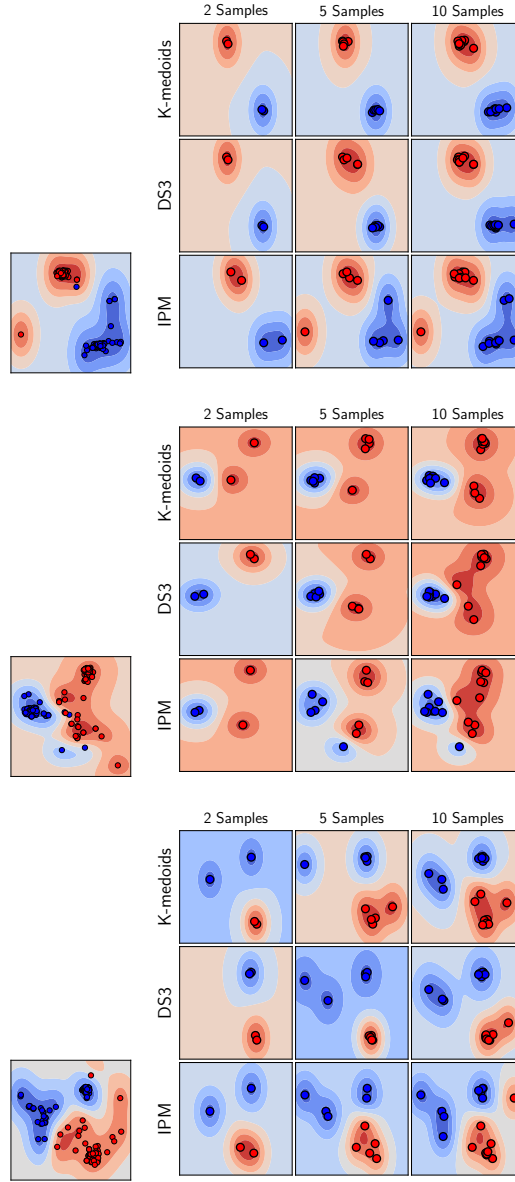


Figure C.2: t-SNE visualization [3] of different randomly selected pairs of classes of UCF-101 dataset and their representatives selected by different methods. (Left) Decision function learned by using all the data. The goal of selection is to preserve the structure with only a few representatives. (Right) Decision function learned by using 2 (first column), 5 (second column), and 10 (third column) representatives per class, using K-medoids (first row), DS3 [2] (second row), and IPM (third row). IPM can capture the structure of the data better using the same number of selected samples.

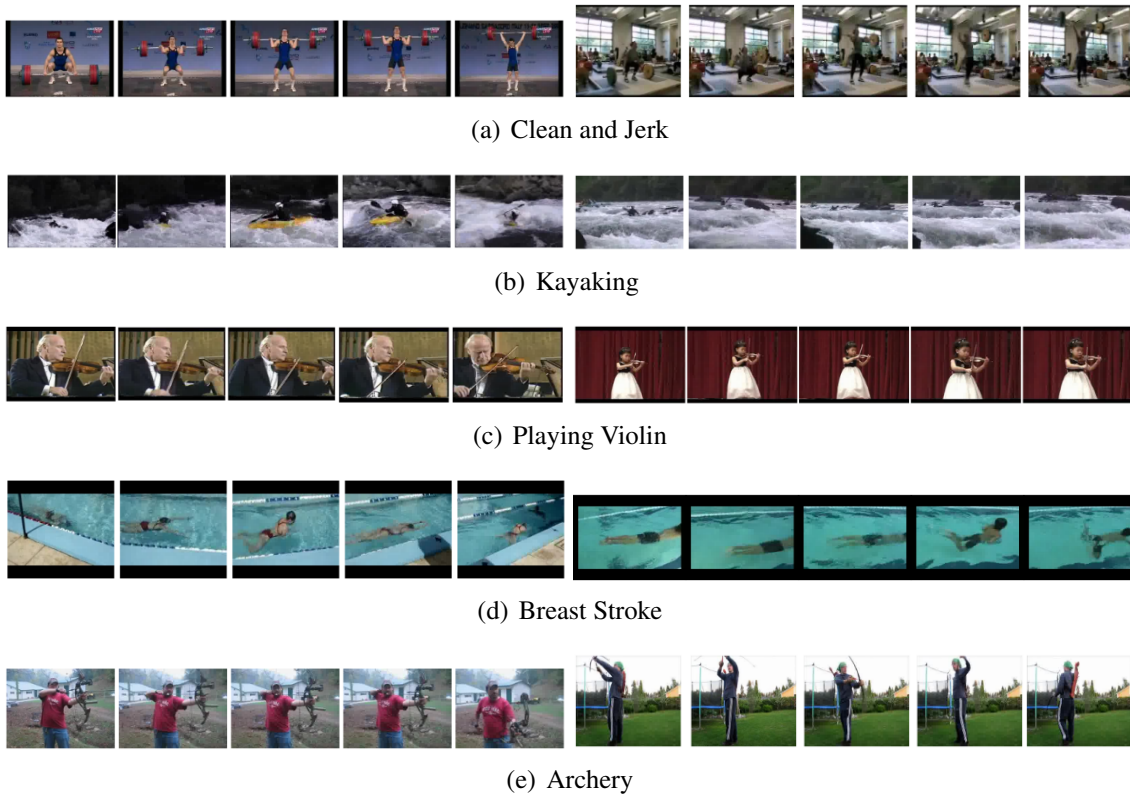


Figure C.3: Frames of the selected video clips by IPM (left) and DS3[2] (right), for a few sample classes of UCF-101 dataset[6]. Different actions are more visible and/or less cluttered, in the clip selected by IPM.

**APPENDIX D: IMPLEMENTATION DETAILS AND FURTHER
EXPERIMENTAL RESULTS FOR THE METHOD PROPOSED IN
CHAPTER 6**

D.1 Implementation Details

Query generation: For CelebA [127] dataset, the standard testing set is used to both generate the queries and as the gallery set. For the synthetic dataset, the latent space of the StyleGAN [109] is sampled to produce the 100,000 images. In Chapter 6, the synthetic images are used for the qualitative evaluations, as the gallery set is much larger.

To generate the queries, we randomly select 1000 images from the gallery set as the query face. We make sure that, after changing one attribute in these query images (the query attribute), there is at least one *similar* image the gallery set. Here, we use the ground truth attributes to define *similarity*. For our experiments, we consider two images similar if they have the exact same ground truth attribute values. Then, we use the query face and the query attribute to create either the modification vector (used by the GAN-based methods) or the modification text (used by the compositional leaning methods). Out of 40 attributes in the CelebA data, 5 attributes are not related to facial features and are removed. These attributes are: `blurry`, `necktie`, `earrings`, `hat`, and `necklace`. Furthermore, to generate modification text and to generate queries, the attributes that describe the same feature are considered as one attribute. For example, CelebA contains ground truth for *black hair*, *brown hair*, *blonde hair*, and *grey hair*. We consider these four attribute as one, when generating the queries. Here are some example query modification texts: `add/remove eyeglasses`, `make hair black/brown/blonde/grey`, `make face young/old`, `add/remove hair`, and `change gender to male/female`.

To generate the modification vector for our method, we just set the corresponding entry to 0 or 1. We use binary modification vector in our experiments for a fair comparison with the text based methods. However, our method is capable of accepting any value between 0 and 1 for the modification vector, which will be illustrated shortly.

To retrieve images using the method in [120], we first use the image embedder to embed the query face. Then, the attribute operator corresponding to the attribute being adjusted is applied to obtain the modified query. The closest faces to this modified query vector in the gallery set are then retrieved and sorted using their Euclidean distance. For the feature extractor, which is a building block of the image embedder architecture in [120], we use Inception Resnet V1 architecture, as described in [129] and trained on VGGFace2 [130].

Training: For the compositional learning baselines, the full training set is used. For each training image, we generate all the possible query modification texts, as discussed earlier. All these possible queries are used to train the model using the code provided by the authors. On the other hand, for our method, we use a subset of CelebA training set and its corresponding attribute ground truth to obtain the attribute direction in a pretrained StyleGAN. For that, we first select a subset of images such that we have both positive and negative for all the attributes. Then, the selected samples are encoded onto the latent space using the encoder proposed in [4]. Then the latent vectors are used to obtain the sparse and orthogonal attribute directions as proposed in Chapter 6. The same number of samples and same encoder are used to extract the attribute directions for the GAN-based baseline [8], using the code provided by the authors.

D.2 Additional Experiments

Figure D.1 illustrates a retrieval example using synthetic images and real-valued modification vector, as opposed to binary. In this example, the user is modifying the attribute `Pale Skin`. The estimated intensity of this attribute in the query is 0.12, but the user is able to modify the retrieval results by increasing it to 0.5 or 1. Here, we have first emphasized this attribute in the results, by increasing the preference value, to make the changes in attribute intensity more dominant. This example shows how our method can successfully utilize a modification vector to manipulate the

results in a continuous manner, a capability which modification text cannot provide.

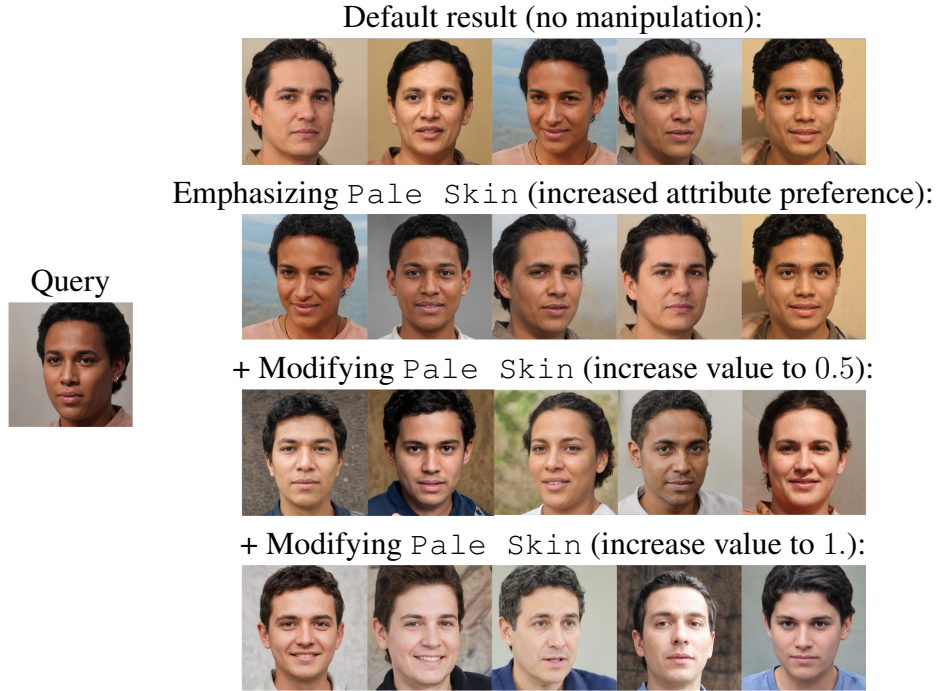


Figure D.1: An example of modifying the retrieval results using continuous, real-valued, modification vector. The attribute intensity for `Pale Skin` for the query face is estimated as 0.12. The user is able to modify the results by increasing it to 0.5 and then to 1.

To compare the retrieved images using our method and the baseline in [7], Figure D.2 shows a few examples of retrieved images and their corresponding performance metrics using the CelebA dataset and after modifying an attribute. For a fair comparison with the text-based baseline, we only use binary values as the modification for this experiment. For example, in Figure D.2(a), we want to retrieve images similar to the query images, while changing the value for attribute `Young` to 0. Note that our method is able to preserve most the other attributes, such as skin tone, hair color, makeup, smiling, etc, while being able to modify the specified attribute, i.e. age. Similarly, for the other examples, the retrieved images by our method are more similar to the query images and to the other retrieved images, both in terms of identity and facial attributes. We argue that this

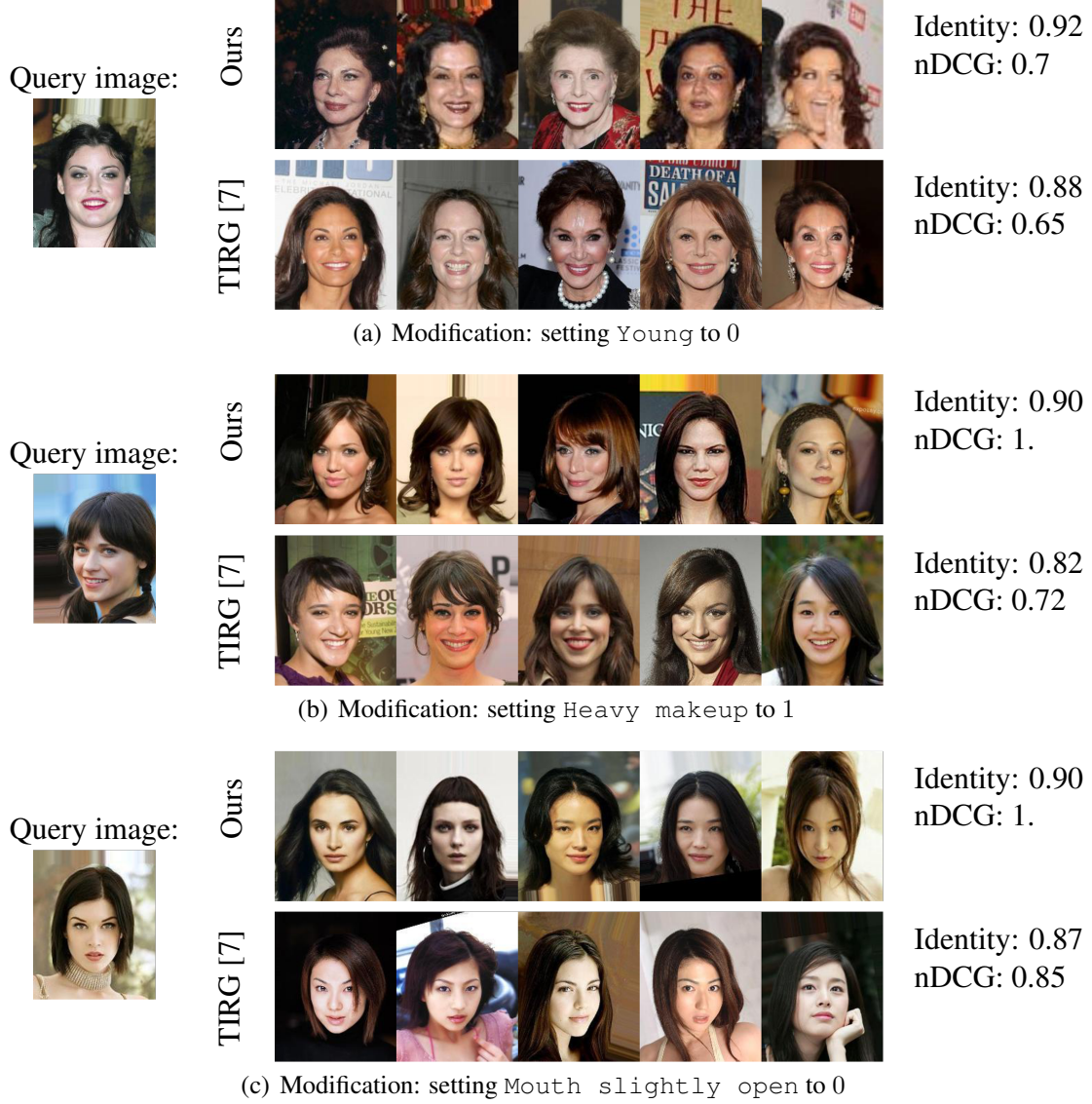


Figure D.2: Examples of retrieved images by our method and the compositional learning method in [7] and their corresponding nDCG and identity similarity. (a) Changing the attribute Young to 0, (b) Changing the attribute Heavy makeup to 1, and (c) Changing the attribute Mouth slightly open to 0. In all of these examples, our method outperforms the baseline in both the evaluation metrics. Qualitatively, the retrieved images by method can modify the attribute, while preserving the other attributes, such as skin tone, hair color, smiling, etc, better.

because the latent space of GAN contains all the necessary information necessary to reconstruct the image, while the embedding space generated by the compositional learning methods does not need to satisfy such requirement. Also, our method is able to disentangle the attributes more effectively and can modify an attribute, while preserving other attributes and the identity.

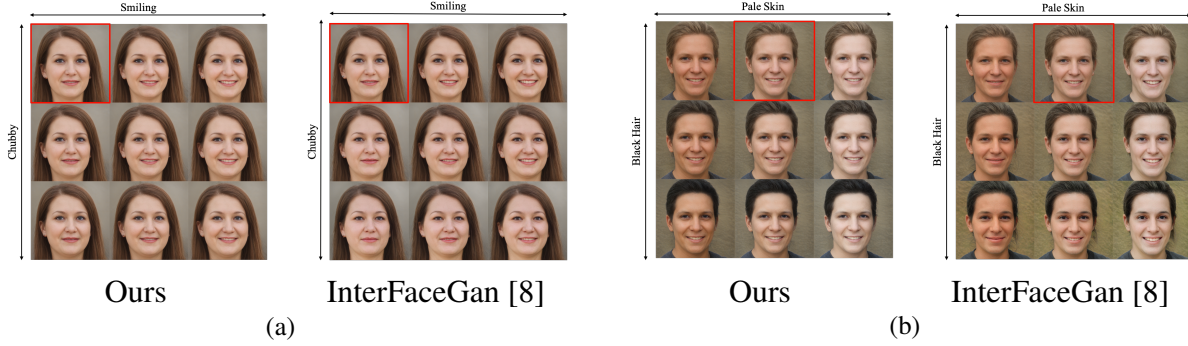


Figure D.3: Attribute manipulation results using our method and the method proposed in [8] on two synthetic images. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. Notice the impact of adjusting attributes Chubby and Pale skin on the smile in images edited using [8].

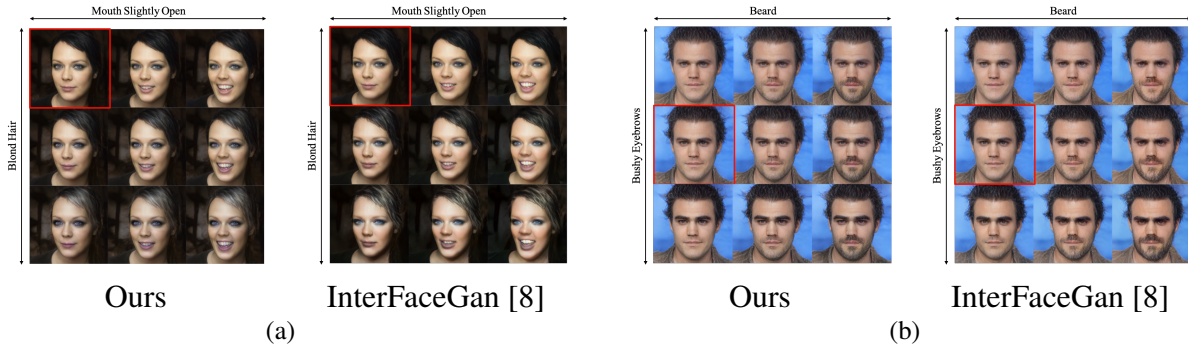


Figure D.4: Attribute manipulation results using our method and the method proposed in [8] on two images from CelebA dataset. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. The obtained directions by the baseline leads to more artifacts compared to the directions obtained by our method.

To illustrate this, Figure D.3 and Figure D.4 show a few examples of editing multiple attributes in

faces using the obtained attribute directions for synthetic and real faces, respectively. To achieve this, the latent vector corresponding to the starting point face, marked with the red square, is moved along two attribute directions. these figures show that our obtained attribute directions are more disentangled, compared to the method proposed in [8]. For example, in Figure D.3, attributes `Pale Skin` and `Chubby` affect the attribute `Smiling` in faces edited using the baseline directions, an artifact that is not present in faces edited by our obtained directions. Furthermore, in Figure D.3(b), manipulating the attribute `Black Hair` using the method in [8] affect the identity. The difference is even more apparent for real faces, Figure D.4, where the baseline modifications lead to a lot more artifacts and more impact on the identity, compared to ours.

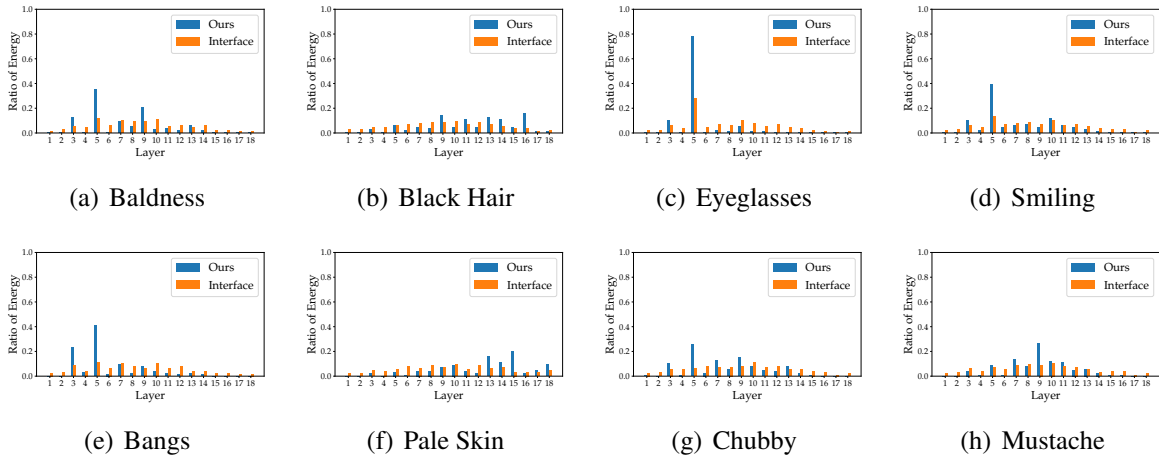


Figure D.5: Ratio of ℓ_2 norm of different attribute vectors in each layer over the total ℓ_2 norm of the vector, for our method and InterFaceGAN [8]. Our method often concentrates most of the energy of the vector in a few layers. For example, vectors corresponding to `Bangs` and `Baldness` have a similar energy profile and only manipulated the layers corresponding to the coarse structures, i.e., first few layers. On the other hand, vectors corresponding to `Black hair` and `Pale skin` mainly change the last few layers, which are responsible for finer structures in the face.

The quantitative results presented in Table 6.1 in Chapter 6 also suggest that the directions obtained by our method are more disentangled compared to [8], as our method is able to consistently achieve better nDCG, while having similar or better identity similarity. This means that our sparse attribute

directions affect the identity and other attributes less. We argue that this is due to the fact that the direction obtained by our method are sparse, meaning that they affect as few entries in the latent vector as possible. This encourages the learned directions to only affect the entries that are most relevant to their corresponding attribute. Figure 6.6 in Chapter 6 shows how the energy of the sparse attribute directions are concentrated on a small percentage of the entries. Similarly, Figure D.5 in this document shows how the energy of the attribute vector is distributed across the layers. The energy ratio for each layer is calculated as the ratio of the ℓ_2 norm of the latent vector in that layer to the overall norm, i.e., $\frac{\|w_l\|_2}{\|w^+\|_2}$. For example, the energy of the attribute vectors that only affect color of skin or hair, `Black Hair` and `Pale Skin`, is mostly concentrated in the layers that are responsible for fine features of the face, i.e., the last few layers of the synthesis network. On the other hand the attribute vectors that affect the coarse structures in the face, such as `Eyeglasses`, `Bangs`, `Baldness`, `Smiling`, etc, are mainly concentrated in the first few layers.

Table D.1: Normalized discounted cumulative gain (nDCG) and identity similarity for the GAN-based methods using different number of training faces to obtain the attribute directions, averaged over 1000 queries. Here we are calculating the metrics on the top-5 images

Number of training samples	3,500		14,000		20,000	
Method	nDCG	Identity Similarity	nDCG	Identity Similarity	nDCG	Identity Similarity
InterFaceGAN [8] (Identity constrained)	0.79	0.817	0.81	0.860	0.82	0.859
Ours (Identity constrained)	0.82	0.830	0.83	0.864	0.85	0.864
InterFaceGAN [8] (best nDCG)	0.83	0.831	0.88	0.839	0.90	0.841
Ours (best nDCG)	0.85	0.840	0.90	0.847	0.92	0.848

Finally, Table D.1 compares the GAN-based methods’ performance, in terms of nDCG and identity similarity, for different number of training samples used to obtain the attribute directions. Our proposed method is consistently more data-efficient compared to the baseline. This can be due to the fact that we enforce both orthogonality and sparsity constraints during the training, which

makes the solution space much smaller. Also, comparing the results with Table 6.1 in Chapter 6, our proposed method can compete with the compositional-learning methods even with only 3,500 training samples, while these baselines use the full training set, containing more than 160,000 samples.

**APPENDIX E: DERIVATION OF UPDATE RULES FOR THE
ALGORITHM PRESENTED IN CHAPTER 9**

In this chapter, the derivations of the update rules for the inference algorithm are presented. As discussed in Section 9.3, the posterior distribution is approximated by a family of distributions for which the calculations are tractable, employing the *naive mean field* approach [146].

In (9.2), \mathcal{H} is divided into disjoint groups $\mathcal{H}_k, k = 1, \dots$, where each \mathcal{H}_k is representing one of the hidden variables in \mathcal{H} . The variational distribution of each partition $\mathbb{Q}\{\mathcal{H}_k\}$ is given by [149, Chapter 10]

$$\ln(\mathbb{Q}\{\mathcal{H}_k\}) = \mathbb{E}_{j \neq k} \{\ln(\mathbb{P}\{\mathcal{D}, \mathcal{H}\})\} + \text{const}, \quad (\text{E.1})$$

where $\mathbb{E}_{j \neq k} \{\cdot\}$ is the expectation with respect to distributions $\mathbb{Q}\{\mathcal{H}_j\}$. Plugging in $\mathbb{P}\{\mathcal{D}, \mathcal{H}\}$ and using the exponential form of the distributions, we obtain the variational distributions. The constant is determined by normalizing the distribution.

It is worthwhile to state that if $x \sim \text{Bernoulli}(p)$, then

$$\ln(\mathbb{P}\{x\}) = \ln\left(\frac{p}{1-p}\right)x + \ln(1-p) \quad (\text{E.2})$$

and if $x \sim \text{Binomial}(n, p)$, we have

$$\ln(\mathbb{P}\{x\}) = \ln\left(\frac{p}{1-p}\right)x + n \ln(1-p) + \ln\left(\binom{n}{x}\right). \quad (\text{E.3})$$

Also, if $x \sim \text{Beta}(b^1, b^0)$, we have

$$\begin{aligned} \ln(\mathbb{P}\{x\}) &= (b^1 - 1) \ln(x) + (b^0 - 1) \ln(1-x) + \text{const} \\ \mathbb{E}\{\ln(x)\} &= \psi(b^1) - \psi(b^1 + b^0), \\ \mathbb{E}\{\ln(1-x)\} &= \psi(b^0) - \psi(b^1 + b^0), \end{aligned} \quad (\text{E.4})$$

where $\psi(\cdot)$ is the digamma function. We now present the update rules to obtain the approximate

posterior distributions.

E.1 Tally Score

Using (E.1), (E.2), (E.4) and integrating out all variables but ϕ_n , we have

$$\begin{aligned}
\ln(\mathbb{Q}\{\phi_n\}) &= \mathbb{E}\{\ln(\mathbb{P}\{\mathcal{D}, \mathcal{H}\})\} + \text{const.} \\
&= \text{const} + \ln(\mathbb{P}\{\phi_n|a_n^1, a_n^0\}) + \mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{\ln(\mathbb{P}\{o_{ni}|u_{ni}, \phi_n\})\} \\
&= \text{const} + (a_n^1 - 1) \ln(\phi_n) + (a_n^0 - 1) \ln(1 - \phi_n) \\
&\quad + \mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\} \left(\ln\left(\frac{\phi_n}{1 - \phi_n}\right) o_{ni} + \ln(1 - \phi_n) \right)
\end{aligned}$$

where i is the updating processor index. The prior knowledge on the tally and (E.4) provide the first two terms; the last combines processor information, given the observation reliability.

This expression can be further written in the form of $(\hat{a}_n^1 - 1) \ln(\phi_n) + (\hat{a}_n^0 - 1) \ln(1 - \phi_n) + \text{const}$, which is a Beta distribution with parameters

$$\begin{aligned}
\hat{a}_n^1 &= a_n^1 + \mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\} o_{ni}, \\
\hat{a}_n^0 &= a_n^0 + \mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\} (1 - o_{ni}).
\end{aligned} \tag{E.5}$$

Here, $\mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{\cdot\}$ is expectation with respect to $\mathbb{Q}\{u_{ni}\}$ and $\mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\}$ can be calculated using $\mathbb{Q}\{u_{ni}\}$, which will be discussed shortly. This update rule simply means that if $o_{ni} = 1$, we will increase the positive count by $\mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\}$; if $o_{ni} = 0$, we will increase the negative count by $\mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\}$.

E.2 Processor Reliability Score

Similarly, to update reliability of each processor i , we have

$$\begin{aligned}
\ln(\mathbb{Q}\{r_i\}) &= \text{const} + (\beta_i^1 - 1) \ln(r_i) + (\beta_i^0 - 1) \ln(1 - r_i) \\
&\quad + \ln\left(\frac{r_i}{1 - r_i}\right) k_i + \ln(1 - r_i) K_i \\
&\quad + \sum_n \mathbb{E}_{\mathbb{Q}\{u_{ni}\}} \left\{ \ln\left(\frac{r_i}{1 - r_i}\right) u_{ni} + \ln(1 - r_i) \right\} \\
&= \text{const} + \ln(r_i) (\beta_i^1 + \sum_n \mathbb{E}_{\mathbb{Q}\{u_{ni}\}} \{u_{ni}\} + k_i - 1) \\
&\quad + \ln(1 - r_i) (\beta_i^0 + \sum_n [1 - \mathbb{E}_{\mathbb{Q}\{u_{ni}\}} \{u_{ni}\}] + K_i - k_i - 1).
\end{aligned}$$

Comparing this to the exponential form of the Beta distribution, we see that $\mathbb{Q}\{r_i\}$ is a Beta distribution with parameters

$$\begin{aligned}
\hat{\beta}_i^1 &= \beta_i^1 + \sum_n \mathbb{E}_{\mathbb{Q}\{u_{ni}\}} \{u_{ni}\} + k_i \\
\hat{\beta}_i^0 &= \beta_i^0 + \sum_n [1 - \mathbb{E}_{\mathbb{Q}\{u_{ni}\}} \{u_{ni}\}] + K_i - k_i.
\end{aligned} \tag{E.6}$$

The sum is only over coefficients with a new observation.

E.3 Observation Reliability

Again, by integrating out all the variables except u_{ni} , we have

$$\begin{aligned}
\ln(\mathbb{Q}\{u_{ni}\}) &= \text{const} + \mathbb{E}_{\mathbb{Q}\{r_i\}} \{ \ln(\mathbb{P}\{u_{ni}|r_i\}) \} \\
&\quad + \mathbb{E}_{\mathbb{Q}\{\phi_n\}} \{ \ln(\mathbb{P}\{o_{ni}|u_{ni}, \phi_n\}) \}.
\end{aligned}$$

By employing (E.2) and (E.4), we have

$$\begin{aligned}
\ln(\mathbb{Q}\{u_{ni}\}) &= u_{ni}\mathbb{E}_{\mathbb{Q}\{r_i\}}\{\ln(r_i)\} + (1 - u_{ni})\mathbb{E}_{\mathbb{Q}\{r_i\}}\{\ln(1 - r_i)\} \\
&+ (1 - u_{ni})\ln(0.5) + u_{ni}[o_{ni}\mathbb{E}_{\mathbb{Q}\{\phi_n\}}\{\ln(\phi_n)\} \\
&+ (1 - o_{ni})\mathbb{E}_{\mathbb{Q}\{\phi_n\}}\{\ln(1 - \phi_n)\}] + \text{const.}
\end{aligned}$$

This update rule, like the others, is a simple expression, as the observations are either 0 or 1. The inference algorithm cannot update u_{ni} if processor i has not reported a measurement on coefficient n . Thus, the update rule is employed for each coefficient on which processor i has a new observation.

To update the distribution, we evaluate the expression for $u_{ni} = 0$ and $u_{ni} = 1$. Since $\mathbb{Q}\{\phi_n\}$ and $\mathbb{Q}\{r_i\}$ are Beta distributions, $\mathbb{E}_{\mathbb{Q}\{\phi_n\}}\{\ln(\phi_n)\}$, $\mathbb{E}_{\mathbb{Q}\{\phi_n\}}\{\ln(1 - \phi_n)\}$, $\mathbb{E}_{\mathbb{Q}\{r_i\}}\{\ln(r_i)\}$, and $\mathbb{E}_{\mathbb{Q}\{r_i\}}\{\ln(1 - r_i)\}$ can be calculated using (E.4).

After normalizing the probabilities to have a valid Bernoulli distribution, the parameter of the distribution can be updated as $\tau_{ni} = \mathbb{E}_{\mathbb{Q}\{u_{ni}\}}\{u_{ni}\} = \mathbb{Q}\{u_{ni} = 1\}$.

**APPENDIX F: CHOOSING HYPERPARAMETER FOR ALGORITHM
PROPOSED IN CHAPTER 10**

Here, we establish a connection between the objective function introduced in (10.5) with Huber norm and use the results in robust statistics to tune ϵ . The first summation in (10.5) can be seen as an iterative approximation of $\sum_i \rho_\epsilon(e_i)$, where

$$\rho_\epsilon(x) = \frac{x^2}{x^2 + \epsilon^2}. \quad (\text{F.1})$$

$\rho_\epsilon(\cdot)$ indicates a measurement as an outlier if the residual is greater than a threshold and this threshold is a function of ϵ . Robustness to noise is improved by increasing the value of ϵ , at the expense of losing robustness to the outlier measurements. Hence, as the variance of noise increases, we should assign a larger ϵ to $\rho_\epsilon(\cdot)$. To set the value of ϵ , a link between the proposed problem and the Huber norm is established.

In robust statistics [208], Huber norm, $\rho_\tau^H(\cdot)$, is utilized to disregard the outlier measurements. $\rho_\tau^H(\cdot)$ is defined as

$$\rho_\tau^H(x) = \begin{cases} \frac{1}{2}x^2 & : |x| < \tau \\ \tau|x| - \frac{\tau^2}{2} & : |x| \geq \tau \end{cases} \quad (\text{F.2})$$

Assuming that the additive measurement noise is Gaussian, the estimator would be 95% asymptotically efficient, meets Cràmer-Rao bound, by setting the parameter τ to 1.34σ , where σ^2 is the variance of the noise [208].

The Huber norm is a convex function. To use the results of robust statistics in the proposed problem, a convex version of the cost function in (F.1) should be employed. The function $\rho_\epsilon(\cdot)$, can be surrogated by its closest convex approximation,

$$\rho_\epsilon^c(x) = \begin{cases} \frac{x^2}{x^2 + \epsilon_0^2} & : |x| < \epsilon_0 \\ \frac{1}{8}(\frac{3}{\epsilon_0}|x| - 1) & : |x| \geq \epsilon_0 \end{cases} \quad (\text{F.3})$$

with $\epsilon_0 = \frac{\epsilon}{\sqrt{3}}$. Figure F.1 illustrates the similarity between the Huber norm and the convex approxi-

mation of $\rho_\epsilon(\cdot)$, i.e., $\rho_\epsilon^c(x)$. The cost functions resemble a least square estimator for errors less than a cut-off parameter, which is the optimal cost function for Gaussian noise. On the other hand, for large values of error, the cost functions resemble the ℓ_0 or ℓ_1 norms, which are known to promote sparsity.

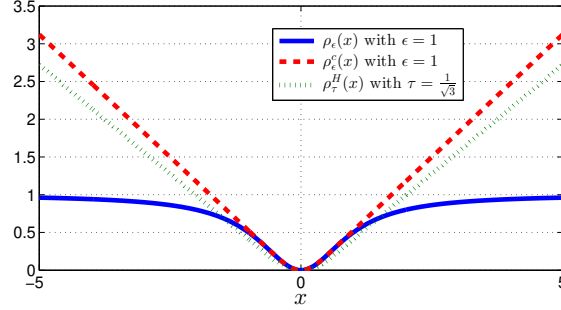


Figure F.1: Comparison of the IRLS weight function, its convex approximation, and the Huber norm.

By extending the results of robust statistics to the proposed problem, we utilize the same cut-off parameter for $\rho_\epsilon^c(x)$ as the Huber norm. It means that for the case of Gaussian noise, we set $\epsilon = 1.34 \sqrt{3} \sigma$, assuming that the nominal noise variance is available. If σ is unknown, an estimation of it can be used [217, Sec. 4.4]. The numerical experiments in Section 10.3 show that the estimator meets the Cràmer-Rao lower bound for sufficiently large number of sensors, by setting $\epsilon = 1.34 \sqrt{3} \sigma$.

APPENDIX G: PROOF OF THEOREM 10.3

Algorithm 6 alternates between two subproblems introduced in (10.9) and (10.11). As discussed in Section 10.2.1, the optimization problem in (10.11) is a GTRS and has a global minimizer for all the iterations. Moreover, $\mathbf{y}^{(k)}$, the global minimizer of (10.11), is obtained by exploiting the conditions in (10.14).

Also the optimization problem in (10.9) is strictly convex and the global minimizer, $\mathbf{w}^{(k)}$, can be calculated using the update rule in (10.10) at each iteration.

Lemma G.1. $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\}$ is non-increasing using the update rules in Algorithm 6, i.e.,

$$\mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w}^{(k+1)}) \leq \mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)}), \forall k = 1, 2, \dots$$

Proof. Using the update rules in Algorithm 6, we have

$$\mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w}^{(k+1)}) \leq \mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w}^{(k)}) \leq \mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)}).$$

The first inequality uses the fact that $\mathbf{w}^{(k+1)}$ is the global minimizer of $\mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w})$. Likewise, the second inequality uses the fact that $\mathbf{y}^{(k+1)}$ is the global minimizer of $\mathcal{J}(\mathbf{y}, \mathbf{w}^{(k)})$. \square

Since $\mathcal{J}(\mathbf{y}^{(1)}, \mathbf{w}^{(0)}) < \infty$ and $\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})$ is non-increasing, then either $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\} \rightarrow -\infty$, or $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\}$ converges to some limit and $\{\mathcal{J}(\mathbf{y}^{(k+1)}, \mathbf{w}^{(k+1)}) - \mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\} \rightarrow 0$ as $k \rightarrow \infty$.

Here, by setting the constant $\epsilon > 0$, we can assure that $-\ln w_i > -\infty, \forall i$. Then, it is easy to notice that $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\}$ is bounded and the sequence $\{\mathcal{J}(\mathbf{y}^{(k)}, \mathbf{w}^{(k)})\}$ will converge to a constant value. To study the convergence of the iterates $\{\mathbf{y}^{(k)}, \mathbf{w}^{(k)}\}$, the definition of a limit point is presented [218].

Definition G.1. \bar{x} is a limit point of $\{x^{(k)}\}$ if there exists a subsequence of $\{x^{(k)}\}$ that converges to \bar{x} . Note that every bounded sequence in \mathbb{R}^n has a limit point (or convergent subsequence).

Now there exist a subsequence $\{(\mathbf{y}^{(k_s)}, \mathbf{w}^{(k_s)})\}$ that converges to a limit point $(\mathbf{y}^*, \mathbf{w}^*)$. By plugging in \mathbf{y}^* and \mathbf{w}^* into the update rules, we will have

$$\mathbf{A}^T \mathbf{W}^* (\mathbf{A} \mathbf{y}^* - \mathbf{b}) + \lambda^* (\mathbf{D} \mathbf{y}^* + \mathbf{f}) = 0,$$

$$(\tilde{\mathbf{a}}_i^T \mathbf{y}^* - b_i)^2 + \epsilon^2 - \frac{1}{w_i^*} = 0, \forall i,$$

which are the derivatives of the Lagrange function of (10.6) w.r.t. \mathbf{y} and w_i . Thus, $(\mathbf{y}^*, \mathbf{w}^*)$ is a stationary point of (10.6).

LIST OF REFERENCES

- [1] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas, “CR-GAN: Learning Complete Representations for Multi-view Generation,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, (California), pp. 942–948, International Joint Conferences on Artificial Intelligence Organization, 7 2018.
- [2] E. Elhamifar, G. Sapiro, and S. S. Sastry, “Dissimilarity based sparse subset selection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2182–2197, 2016.
- [3] Laurens van der Maaten, “Visualizing Data using t-SNE,” *Annals of Operations Research*, 2014.
- [4] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, “Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation,” *arXiv preprint arXiv:2008.00951*, 2020.
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 6 2009.
- [6] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” 12 2012.
- [7] N. Vo, L. Jiang, C. Sun, K. Murphy, L. J. Li, L. Fei-Fei, and J. Hays, “Composing text and image for image retrieval-An empirical odyssey,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.

- [8] Y. Shen, J. Gu, X. Tang, and B. Zhou, “Interpreting the latent space of GANs for semantic face editing,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.
- [9] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, pp. 354–359, 10 2017.
- [10] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, “On the Number of Linear Regions of Deep Neural Networks,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2924–2932, Curran Associates, Inc., 2014.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, 6 2016.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, Springer, Cham, 10 2016.
- [13] D. Hendrycks and K. Gimpel, “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks,” *Proceedings of International Conference on Learning Representations*, 2017.
- [14] S. Liang, Y. Li, and R. Srikant, “Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks,” in *International Conference on Learning Representations*, 2018.

- [15] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, “Out-of-distribution detection using an ensemble of self supervised leave-out classifiers,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.
- [16] R. Yehezkel Rohekar, Y. Gurwicz, S. Nisimov, and G. Novik, “Modeling Uncertainty by Learning a Hierarchy of Deep Neural Connections,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 4246–4256, Curran Associates, Inc., 2019.
- [17] A. Bendale and T. E. Boulton, “Towards open set deep networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [18] R. Yoshihashi, S. You, W. Shao, M. Iida, R. Kawakami, and T. Naemura, “Classification-Reconstruction Learning for Open-Set Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [19] K. K. Lee, K. K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Advances in Neural Information Processing Systems*, 2018.
- [20] D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–12, Springer-Verlag New York, Inc., 1994.
- [21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014.

- [22] D. L. D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 4 2006.
- [23] E. E. J. Candès, “Compressive sampling,” 2006.
- [24] S. Salehi, A. Zaeemzadeh, A. Tatulian, N. Rahnavard, and R. R. F. R. DeMara, “MRAM-Based Stochastic Oscillators for Adaptive Non-Uniform Sampling of Sparse Signals in IoT Applications,” in *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, vol. 2019-July, 2019.
- [25] E. J. Candes, T. Tao, E. J. Candès, T. Tao, E. J. Candes, and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, IEEE, 12 2015.
- [27] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 6 2015.
- [28] A. Zaeemzadeh, N. Rahnavard, and M. Shah, “Norm-Preservation: Why Residual Networks Can Become Extremely Deep?,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020.
- [29] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training Very Deep Networks,” 2015.
- [30] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE, 7 2017.

- [31] A. E. Orhan and X. Pitkow, “Skip connections eliminate singularities,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 1 2018.
- [32] M. Hardt and T. Ma, “Identity matters in deep learning,” in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 11 2017.
- [33] K. Kawaguchi, “Deep Learning without Poor Local Minima,” 2016.
- [34] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams, “The Shattered Gradients Problem: If resnets are the answer, then what is the question?,” 7 2017.
- [35] A. Veit, M. J. Wilber, and S. Belongie, “Residual Networks Behave Like Ensembles of Relatively Shallow Networks,” in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 550–558, Curran Associates, Inc., 2016.
- [36] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” 3 2010.
- [37] L. Dinh, J. Sohl-Dickstein Google, B. Samy, and B. Google Brain, “Density estimation using Real NVP,” in *ICLR*, 2017.
- [38] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, “The reversible residual network: Backpropagation without storing activations,” in *Advances in Neural Information Processing Systems*, 2017.
- [39] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J. H. Jacobsen, “Invertible residual networks,” in *36th International Conference on Machine Learning, ICML 2019*, 2019.

- [40] P. L. Bartlett, S. N. Evans, and P. M. Long, “Representing smooth functions as compositions of near-identity functions with implications for deep network optimization,” *arXiv preprint arXiv:1804.05012*, 4 2018.
- [41] K. Kawaguchi and Y. Bengio, “Depth with nonlinearity creates no bad local minima in ResNets,” *Neural Networks*, 2019.
- [42] S. Gunasekar, J. D. Lee, N. Srebro, and D. Soudry, “Implicit bias of gradient descent on linear convolutional networks,” in *Advances in Neural Information Processing Systems*, 2018.
- [43] E. Hoffer, I. Hubara, and D. Soudry, “Train longer, generalize better: Closing the generalization gap in large batch training of neural networks,” in *Advances in Neural Information Processing Systems*, 2017.
- [44] H. Sedghi, V. Gupta, and P. M. Long, “The Singular Values of Convolutional Layers,” in *International Conference on Learning Representations*, 2019.
- [45] J. C. Gower, G. B. Dijkstrahuis, and others, *Procrustes problems*, vol. 30. Oxford University Press on Demand, 2004.
- [46] N. J. Higham, “Stable iterations for the matrix square root,” *Numerical Algorithms*, vol. 15, no. 2, pp. 227–242, 1997.
- [47] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” tech. rep., 2009.
- [48] X. Dong, G. Kang, K. Zhan, and Y. Yang, “EraseReLU: A Simple Way to Ease the Training of Deep Convolution Neural Networks,” *arXiv preprint arXiv:1709.07634*, 2017.
- [49] W. J. Maddox, T. Garipov, Izmailov, D. Vetrov, and A. G. Wilson, “A simple baseline for Bayesian uncertainty in deep learning,” in *Advances in Neural Information Processing Systems*, 2019.

- [50] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” in *PMLR*, 2016.
- [51] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-Distribution Detection Using Union of 1-Dimensional Subspaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [52] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, “Being robust (in high dimensions) can be practical,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.
- [53] M. Joneidi, P. Ahmadi, M. Sadeghi, and N. Rahnavard, “Union of low-rank subspaces detector,” *IET Signal Processing*, 2016.
- [54] K. Ravindran, S. Hadi, and V. Santosh, “The spectral method for general mixture models,” *SIAM Journal on Computing*, 2008.
- [55] A. Zaeemzadeh, M. Joneidi, N. Rahnavard, and M. Shah, “Iterative Projection and Matching: Finding Structure-preserving Representatives and Its Application to Computer Vision,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, 2019.
- [56] B. Tran, J. Li, and A. Madry, “Spectral signatures in backdoor attacks,” in *Advances in Neural Information Processing Systems*, 2018.
- [57] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto, “Generative probabilistic novelty detection with adversarial autoencoders,” in *Advances in Neural Information Processing Systems*, 2018.
- [58] D. Hendrycks, M. Mazeika, and T. Dietterich, “Deep anomaly detection with outlier exposure,” in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

- [59] Q. Yu and K. Aizawa, “Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy,” in *The IEEE International Conference on Computer Vision (ICCV)*, 10 2019.
- [60] I. Golan and R. El-Yaniv, “Deep anomaly detection using geometric transformations,” in *Advances in Neural Information Processing Systems*, 2018.
- [61] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, “Effective End-to-end Unsupervised Outlier Detection via Inlier Priority of Discriminative Network,” *Nips*, 2019.
- [62] K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.
- [63] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, “Likelihood Ratios for Out-of-Distribution Detection,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 14680–14691, Curran Associates, Inc., 2019.
- [64] G. Shalev, Y. Adi, and J. Keshet, “Out-of-distribution detection using multiple semantic label representations,” in *Advances in Neural Information Processing Systems*, 2018.
- [65] L. Neal, M. Olson, X. Fern, W. K. Wong, and F. Li, “Open set learning with counterfactual images,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11210 LNCS, pp. 620–635, 2018.
- [66] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-Scale Long-Tailed Recognition in an Open World,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2019.

- [67] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, “Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data,” 2020.
- [68] M. M. El Ayadi, M. S. Kamel, and F. Karray, “Toward a tight upper bound for the error probability of the binary Gaussian classification problem,” *Pattern Recognition*, 2008.
- [69] D. Chen, T. Zheng, and H. Yang, “Estimates of the gaps between consecutive eigenvalues of Laplacian,” *Pacific Journal of Mathematics*, 2016.
- [70] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large Margin Cosine Loss for Deep Face Recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [71] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [72] A. M. Saxe, J. L. McClelland, and S. Ganguli, “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks,” *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2014.
- [73] P. Comon and G. H. Golub, “Tracking a few extreme singular values and vectors in signal processing,” *Proceedings of the IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.
- [74] J. Baglama and L. Reichel, “Augmented implicitly restarted lanczos bidiagonalization methods,” *SIAM Journal on Scientific Computing*, 2005.
- [75] S. Liu, R. Garrepalli, T. G. Dietterich, A. Fern, and D. Hendrycks, “Open category detection with PAC guarantees,” in *35th International Conference on Machine Learning, ICML 2018*, 2018.

- [76] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop,” 2015.
- [77] S. Zagoruyko and N. Komodakis, “Wide Residual Networks,” in *British Machine Vision Conference 2016, BMVC 2016*, 2016.
- [78] K. Hara, H. Kataoka, and Y. Satoh, “Learning spatio-Temporal features with 3D residual networks for action recognition,” in *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2017.
- [79] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in Neural Information Processing Systems*, 2017.
- [80] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak, “On mixup training: Improved calibration and predictive uncertainty for deep neural networks,” in *Advances in Neural Information Processing Systems*, 2019.
- [81] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.
- [82] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [83] S. Joshi and S. Boyd, “Sensor Selection via Convex Optimization,” *IEEE Transactions on Signal Processing*, vol. 57, pp. 451–462, 2 2009.
- [84] M. Shamaiah, S. Banerjee, and H. Vikalo, “Greedy sensor selection: Leveraging submodularity,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 2572–2577, 12 2010.

- [85] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600–1607, IEEE, 2012.
- [86] Y. Gal, R. Islam, and Z. Ghahramani, “Deep Bayesian Active Learning with Image Data,” in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 1183–1192, PMLR, 2017.
- [87] K. Hara, H. Kataoka, and Y. Satoh, “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Salt Lake City, UT), pp. 6546–6555, 2018.
- [88] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The Kinetics Human Action Video Dataset,” 5 2017.
- [89] J. A. Tropp and A. C. Gilbert, “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit,” *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, 12 2007.
- [90] T. T. Cai and L. Wang, “Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise,” *Information Theory, IEEE Transactions on*, vol. 57, pp. 4680–4688, 7 2011.
- [91] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-PIE,” *Image and Vision Computing*, vol. 28, pp. 807–813, 5 2010.
- [92] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “MS-Celeb-1M: Challenge of Recognizing One Million Celebrities in the Real World,” *Electronic Imaging*, vol. 2016, pp. 1–6, 2 2016.

- [93] Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin, “Unsupervised Feature Learning via Non-Parametric Instance Discrimination,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [94] Yong Jae Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, IEEE, 6 2012.
- [95] Z. Lu and K. Grauman, “Story-Driven Summarization for Egocentric Video,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2714–2721, IEEE, 6 2013.
- [96] S. Yeung, A. Fathi, and L. Fei-Fei, “VideoSET: Video Summary Evaluation through Text,” 6 2014.
- [97] B. A. Plummer, M. Brown, and S. Lazebnik, “Enhancing Video Summarization via Vision-Language Embedding,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1052–1060, IEEE, 7 2017.
- [98] M. Gygli, H. Grabner, and L. Van Gool, “Video summarization by learning submodular mixtures of objectives,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3090–3098, IEEE, 6 2015.
- [99] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries,” *Annual Meeting of the Association for Computational Linguistics*, 2004.
- [100] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, “Diverse Sequential Subset Selection for Supervised Video Summarization,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2069–2077, Curran Associates, Inc., 2014.

- [101] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, IEEE, 6 2015.
- [102] K. Zhang, W. L. Chao, F. Sha, and K. Grauman, “Video summarization with long short-term memory,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.
- [103] X. Yang, X. Song, X. Han, H. Wen, J. Nie, and L. Nie, “Generative Attribute Manipulation Scheme for Flexible Fashion Search,” in *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.
- [104] Z. Yu and A. Kovashka, “Syntharch: Interactive image search with attribute-conditioned synthesis,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020.
- [105] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [106] A. Kovashka, D. Parikh, and K. Grauman, “WhittleSearch: Interactive Image Search with Relative Attribute Feedback,” *International Journal of Computer Vision*, 2015.
- [107] X. Guo, S. Rennie, H. Wu, G. Tesauro, Y. Cheng, and R. S. Feris, “Dialog-based Interactive Image Retrieval,” in *Advances in Neural Information Processing Systems*, 2018.
- [108] B. Zhao, J. Feng, X. Wu, and S. Yan, “Memory-augmented attribute manipulation networks for interactive fashion search,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

- [109] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.
- [110] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.
- [111] X. Wang, H. Zhang, W. Huang, and M. R. Scott, “Cross-Batch Memory for Embedding Learning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.
- [112] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, “Deep metric learning to rank,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.
- [113] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer, “Divide and conquer the embedding space for metric learning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019.
- [114] B. Brattoli, K. Roth, and B. Ommer, “MIC: Mining interclass characteristics for improved metric learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
- [115] I. Misra, A. Gupta, and M. Hebert, “From red wine to red tomato: Composition with context,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [116] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis, “Auto-

- matic Spatially-Aware Fashion Concept Discovery,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [117] K. E. Ak, A. A. Kassim, J. H. Lim, and J. Y. Tham, “Learning Attribute Representations with Localization for Flexible Fashion Search,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [118] X. Han, Z. Wu, Y. G. Jiang, and L. S. Davis, “Learning fashion compatibility with bidirectional LSTMs,” in *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, 2017.
- [119] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [120] T. Nagarajan and K. Grauman, “Attributes as Operators,” *European Conference on Computer Vision*, 2018.
- [121] H. Noh, P. H. Seo, and B. Han, “Image question answering using convolutional neural network with dynamic parameter prediction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- [122] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, “A simple neural network module for relational reasoning,” in *Advances in Neural Information Processing Systems*, 2017.
- [123] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018.

- [124] H. Shao, A. Kumar, and P. Thomas Fletcher, “The riemannian geometry of deep generative models,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018.
- [125] S. Laine, “Feature-based metrics for exploring the latent space of generative models,” in *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018.
- [126] A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H. P. Seidel, P. Perez, M. Zollhofer, and C. Theobalt, “Stylerig: Rigging stylegan for 3d control over portrait images,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.
- [127] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [128] N. Boumal, P. A. Absil, and C. Cartis, “Global rates of convergence for nonconvex optimization on manifolds,” *IMA Journal of Numerical Analysis*, 2019.
- [129] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.
- [130] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGFace2: A dataset for recognising faces across pose and age,” in *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, 2018.
- [131] A. Zaeemzadeh, M. Joneidi, and N. Rahnavard, “Adaptive non-uniform compressive sampling for time-varying signals,” in *51st Annual Conference on Information Sciences and Systems, CISS 2017*, 2017.

- [132] B. Shahrasbi and N. Rahnavard, "Model-Based Nonuniform Compressive Sampling and Recovery of Natural Images Utilizing a Wavelet-Domain Universal Hidden Markov Model," *IEEE Transactions on Signal Processing*, 2017.
- [133] M. Y. S. Uddin, H. Wang, F. Saremi, G.-J. Qi, T. Abdelzaher, and T. Huang, "PhotoNet: A Similarity-Aware Picture Delivery Service for Situation Awareness," in *2011 IEEE 32nd Real-Time Systems Symposium*, pp. 317–326, IEEE, 11 2011.
- [134] A. Rahimpour, A. Taalimi, J. Luo, and H. Qi, "Distributed object recognition in smart camera networks," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 669–673, IEEE, 9 2016.
- [135] M. Leinonen, M. Codreanu, and M. Juntti, "Sequential Compressed Sensing With Progressive Signal Reconstruction in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 1622–1635, 3 2015.
- [136] A. Sani and A. Vosoughi, "Distributed Vector Estimation for Power- and Bandwidth-Constrained Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 64, pp. 3879–3894, 8 2016.
- [137] M. Rahmani and G. Atia, "A Subspace Learning Approach for High Dimensional Matrix Decomposition with Efficient Column/Row Sampling," 2016.
- [138] M.-p. Hosseini, H. Soltanian-zadeh, K. Elisevich, and D. Pompili, "Cloud-based Deep Learning of Big EEG Data for Epileptic Seizure Prediction," in *Proc. of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, (Greater Washington, D.C., USA), pp. 5–9, IEEE, 12 2016.
- [139] A. Fragkiadakis, P. Charalampidis, and E. Tragos, "Adaptive compressive sensing for energy efficient smart objects in IoT applications," in *2014 4th International Conference on Wire-*

- less Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, pp. 1–5, IEEE, 5 2014.
- [140] J. Ziniel and P. Schniter, “Dynamic Compressive Sensing of Time-Varying Signals Via Approximate Message Passing,” *IEEE Transactions on Signal Processing*, vol. 61, pp. 5270–5284, 11 2013.
- [141] U. L. Wijewardhana and M. Codreanu, “A Bayesian Approach for Online Recovery of Streaming Signals from Compressive Measurements,” *IEEE Transactions on Signal Processing*, pp. 1–1, 2016.
- [142] B. Shahrasbi, A. Talari, and N. Rahnavard, “TC-CSBP: Compressive sensing for time-correlated data based on belief propagation,” in *Annual Conference on Information Sciences and Systems*, pp. 1–6, IEEE, 3 2011.
- [143] M. L. Malloy and R. D. Nowak, “Near-Optimal Adaptive Compressed Sensing,” *IEEE Transactions on Information Theory*, vol. 60, pp. 4001–4012, 7 2014.
- [144] G. Braun, S. Pokutta, and Y. Xie, “Info-Greedy Sequential Adaptive Compressed Sensing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, pp. 601–611, 6 2015.
- [145] J. Haupt, R. Baraniuk, R. Castro, and R. Nowak, “Sequentially designed compressed sensing,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 401–404, IEEE, 8 2012.
- [146] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends[®] in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.
- [147] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, “An introduction to variational methods for graphical models,” *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

- [148] B. Babagholami-Mohamadabadi, A. Jourabloo, A. Zarghami, and S. Kasaei, "A Bayesian Framework for Sparse Representation-Based 3-D Human Pose Estimation," *IEEE Signal Processing Letters*, vol. 21, pp. 297–300, 3 2014.
- [149] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [150] M. A. Iwen and A. H. Tewfik, "Adaptive Strategies for Target Detection and Localization in Noisy Environments," *IEEE Transactions on Signal Processing*, vol. 60, pp. 2344–2353, 5 2012.
- [151] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008.
- [152] M. Grant and S. Boyd, "{CVX}: Matlab Software for Disciplined Convex Programming, version 1.21," 2 2011.
- [153] Q. Zhao, "A survey of dynamic spectrum access: signal processing, networking, and regulatory policy," in *IEEE Signal Processing Magazine*, pp. 79–89, 2007.
- [154] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [155] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE journal on selected areas in communications*, vol. 23, no. 2, pp. 201–220, 2005.
- [156] A. Zaeemzadeh, M. Joneidi, B. Shahrabi, and N. Rahnavard, "Missing spectrum-data recovery in cognitive radio networks using piecewise constant Nonnegative Matrix Factorization," in *Proceedings - IEEE Military Communications Conference MILCOM*, vol. 2015-Decem, pp. 238–243, IEEE, 10 2015.

- [157] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 4, pp. 40–62, 12 2011.
- [158] B. Shahrabi, N. Rahnavard, and A. Vosoughi, "Cluster-CMSS: A Cluster-Based Coordinated Spectrum Sensing in Geographically Dispersed Mobile Cognitive Radio Networks," *IEEE Transactions on Vehicular Technology*, vol. pp, no. pp, pp. 1–1, 2016.
- [159] O. P. Awe, Z. Zhu, and S. Lambotharan, "Eigenvalue and Support Vector Machine Techniques for Spectrum Sensing in Cognitive Radio Networks," in *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on*, pp. 223–227, 12 2013.
- [160] S. Amini, M. Sadeghi, M. Joneidi, M. Babaie-Zadeh, and C. Jutten, "Outlier-aware dictionary learning for sparse representation," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, pp. 1–6, 9 2014.
- [161] S.-J. Kim and G. B. Giannakis, "Cognitive radio spectrum prediction using dictionary learning," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 3206–3211, 12 2013.
- [162] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
- [163] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.
- [164] S. Essid and C. Fevotte, "Smooth Nonnegative Matrix Factorization for Unsupervised Audiovisual Document Structuring," *Multimedia, IEEE Transactions on*, vol. 15, pp. 415–425, 2 2013.

- [165] Z. Hu, R. Ranganathan, C. Zhang, R. Qiu, M. Bryant, M. C. Wick, and L. Li, “Robust non-negative matrix factorization for joint spectrum sensing and primary user localization in cognitive radio networks,” in *IEEE Waveform Diversity and Design Conference*, 2012.
- [166] N. Seichepine, S. Essid, C. Fevotte, and O. Cappe, “Piecewise constant nonnegative matrix factorization,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 6721–6725, 5 2014.
- [167] X. Fu, N. D. Sidiropoulos, and W.-K. Ma, “Tensor-based power spectra separation and emitter localization for cognitive radio,” in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*, pp. 421–424, 6 2014.
- [168] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, pp. 556–562, 2001.
- [169] C. Févotte and J. Idier, “Algorithms for nonnegative matrix factorization with the β -divergence,” *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.
- [170] V. Y. Tan and C. Févotte, “Automatic relevance determination in nonnegative matrix factorization with the (β) -divergence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1592–1605, 7 2013.
- [171] J. M. Bioucas-Dias, M. Figueiredo, J. P. Oliveira, and others, “Adaptive total variation image deconvolution: A majorization-minimization approach,” in *Signal Processing Conference, 2006 14th European*, pp. 1–4, IEEE, 2006.
- [172] S. Minaee and Y. Wang, “Screen Content Image Segmentation Using Least Absolute Deviation Fitting,” *arXiv preprint arXiv:1501.03755*, 2015.
- [173] Y. Mao and L. K. Saul, “Modeling Distances in Large-scale Networks by Matrix Factor-

- ization,” in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, (New York, NY, USA), pp. 278–287, ACM, 2004.
- [174] Y.-D. Kim and S. Choi, “Weighted nonnegative matrix factorization,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1541–1544, 4 2009.
- [175] D. Needell and R. Ward, “Stable Image Reconstruction Using Total Variation Minimization,” *SIAM Journal on Imaging Sciences*, vol. 6, pp. 1035–1058, 1 2013.
- [176] J. Tan, Y. Ma, H. Rueda, D. Baron, and G. R. Arce, “Compressive Hyperspectral Imaging via Approximate Message Passing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, pp. 389–401, 3 2016.
- [177] C. Jeon, R. Ghods, A. Maleki, and C. Studer, “Optimality of large MIMO detection via approximate message passing,” in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1227–1231, IEEE, 6 2015.
- [178] M. Joneidi, A. Zaeemzadeh, B. Shahrabi, G.-J. Qi, and N. Rahnavard, “E-optimal Sensor Selection for Compressive Sensing-based Purposes,” *IEEE Transactions on Big Data*, p. To Appear in, 2018.
- [179] S. Salehi, M. B. Mashhadi, A. Zaeemzadeh, N. Rahnavard, R. F. De Mara, and R. F. De Mara, “Energy-Aware Adaptive Rate and Resolution Sampling of Spectrally Sparse Signals Leveraging VCMA-MTJ Devices,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, 12 2018.
- [180] A. Zaeemzadeh, J. Haddock, N. Rahnavard, and D. Needell, “A Bayesian Approach for Asynchronous Parallel Sparse Recovery,” in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 2018-Octob, pp. 1980–1984, IEEE, 10 2019.

- [181] J. Zhu, R. Pilgrim, and D. Baron, “An overview of multi-processor approximate message passing,” in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, IEEE, 3 2017.
- [182] S. Patterson, Y. C. Eldar, and I. Keidar, “Distributed Compressed Sensing for Static and Time-Varying Networks,” *IEEE Transactions on Signal Processing*, vol. 62, pp. 4931–4946, 10 2014.
- [183] C. Ravazzi, S. M. Fosson, and E. Magli, “Distributed iterative thresholding for ℓ_0/ℓ_1 -regularized linear inverse problems,” *IEEE Transactions on Information Theory*, vol. 61, pp. 2081–2100, April 2015.
- [184] Y. Ma, Y. M. Lu, and D. Baron, “Multiprocessor approximate message passing with column-wise partitioning,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017.
- [185] P. Han, R. Niu, M. Ren, and Y. C. Eldar, “Distributed approximate message passing for sparse signal recovery,” in *2014 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2014*, 2014.
- [186] P. Han, J. Zhu, R. Niu, and D. Baron, “Multi-processor approximate message passing using lossy compression,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016.
- [187] D. Needell and T. Woolf, “An asynchronous parallel approach to sparse recovery,” in *2017 Information Theory and Applications Workshop (ITA)*, pp. 1–5, IEEE, 2 2017.
- [188] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, pp. 265–274, 11 2009.

- [189] N. Nguyen, D. Needell, and T. Woolf, “Linear Convergence of Stochastic Iterative Greedy Algorithms With Sparse Constraints,” *IEEE Transactions on Information Theory*, vol. 63, pp. 6869–6895, 11 2017.
- [190] A. Zaeemzadeh, M. Joneidi, B. Shahrabi, and N. Rahnavard, “Robust Target Localization Based on Squared Range Iterative Reweighted Least Squares,” in *2017 IEEE 14th International Conference on Mobile Adhoc and Sensor Systems*, IEEE Computer Society, 2017.
- [191] L. Dagum and R. Menon, “OpenMP: an industry standard API for shared-memory programming,” *IEEE Computational Science and Engineering*, vol. 5, no. 1, pp. 46–55, 1998.
- [192] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, pp. 18914–18919, 11 2009.
- [193] P. Oguz-Ekim, J. P. Gomes, J. Xavier, and P. Oliveira, “Robust Localization of Nodes and Time-Recursive Tracking in Sensor Networks Using Noisy Range Measurements,” *Signal Processing, IEEE Transactions on*, vol. 59, pp. 3930–3942, 8 2011.
- [194] A. Beck, P. Stoica, and J. Li, “Exact and Approximate Solutions of Source Localization Problems,” *Signal Processing, IEEE Transactions on*, vol. 56, pp. 1770–1778, 5 2008.
- [195] G. Destino and G. Abreu, “On the Maximum Likelihood Approach for Source and Network Localization,” *Signal Processing, IEEE Transactions on*, vol. 59, pp. 4954–4970, 10 2011.
- [196] Y. Jiang and M. R. Azimi-Sadjadi, “A Robust Source Localization Algorithm Applied to Acoustic Sensor Network,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, pp. III–1233–III–1236, 4 2007.
- [197] H. Jamali-Rad and G. Leus, “Sparsity-aware TDOA localization of multiple sources,” in

- Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 4021–4025, 5 2013.
- [198] Y. Zhang, K. Yang, and M. G. Amin, “Robust target localization in moving radar platform through semidefinite relaxation,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 2209–2212, 4 2009.
- [199] G. Wang and K. Yang, “A New Approach to Sensor Node Localization Using RSS Measurements in Wireless Sensor Networks,” *Wireless Communications, IEEE Transactions on*, vol. 10, pp. 1389–1395, 5 2011.
- [200] S. Yousefi, X.-W. Chang, and B. Champagne, “Distributed cooperative localization in wireless sensor networks without NLOS identification,” in *2014 11th Workshop on Positioning, Navigation and Communication (WPNC)*, pp. 1–6, 3 2014.
- [201] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “TOA-based robust wireless geolocation and crammer-rao lower bound analysis in harsh LOS/NLOS environments,” *IEEE Transactions on Signal Processing*, vol. 61, pp. 2243–2255, 5 2013.
- [202] J. J. Mor, “Generalizations Of The Trust Region Problem,” *OPTIMIZATION METHODS AND SOFTWARE*, vol. 2, pp. 189–209, 1993.
- [203] M. Hussain, Y. Aytar, N. Trigoni, and A. Markham, “Characterization of non-line-of-sight (NLOS) bias via analysis of clutter topology,” in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 1247–1256, 4 2012.
- [204] S. Nawaz and N. Trigoni, “Convex programming based robust localization in NLOS prone cluttered environments,” in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pp. 318–329, 4 2011.

- [205] F. Gustafsson and F. Gunnarsson, “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements,” *Signal Processing Magazine, IEEE*, vol. 22, pp. 41–53, 7 2005.
- [206] F. Yin and A. M. Zoubir, “Robust positioning in NLOS environments using nonparametric adaptive kernel density estimation,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3517–3520, 3 2012.
- [207] P.-C. Chen, “A non-line-of-sight error mitigation algorithm in location estimation,” in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pp. 316–320, 1999.
- [208] P. J. Huber, *Robust statistics*. Springer, 2011.
- [209] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.
- [210] P. Pennacchi, “Robust estimate of excitations in mechanical systems using M-estimators – Theoretical background and numerical applications,” *Journal of Sound and Vibration*, vol. 310, no. 4–5, pp. 923–946, 2008.
- [211] S. Geman and D. E. McClure, “Statistical methods for tomographic image reconstruction,” 1987.
- [212] R. Chartrand and W. Yin, “Iteratively reweighted algorithms for compressive sensing,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3869–3872, 3 2008.
- [213] M. Boloursaz Mashhadi, N. Salarieh, E. S. Farahani, and F. Marvasti, “Level crossing

- speech sampling and its sparsity promoting reconstruction using an iterative method with adaptive thresholding,” *IET Signal Processing*, vol. 11, pp. 721–726, 8 2017.
- [214] Y. Xu and W. Yin, “A globally convergent algorithm for nonconvex optimization based on block coordinate update,” *arXiv preprint arXiv:1410.1386*, 2014.
- [215] C. Zhang, B. Recht, S. Bengio, M. Hardt, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.
- [216] N. A. Derzko and A. M. Pfeffer, “Bounds for the Spectral Radius of a Matrix,” *Mathematics of Computation*, vol. 19, p. 62, 4 1965.
- [217] R. Maronna, D. Martin, and V. Yohai, *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.
- [218] M. Razaviyayn, M. Hong, and Z.-Q. Luo, “A Unified Convergence Analysis of Block Successive Minimization Methods for Nonsmooth Optimization,” *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.